SINGLE-CHANNEL SPEECH ENHANCEMENT BASED ON DEEP NEURAL NETWORKS

Zhiheng Ouyang

A thesis

in

The Department

of

Electrical and computer engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science

Concordia University

Montréal, Québec, Canada

December 2019

© Zhiheng Ouyang, 2020

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Zhiheng Ouyang

Entitled: Single-Channel Speech Enhancement Based on Deep Neural Networks

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. H. Rivaz

_____ External Examiner
Dr. C.-Y. Su (MIAE)

_____ Internal Examiner
Dr. H. Rivaz

_____ Supervisor
Dr. W.-P. Zhu

Approved by: _____
Dr. Y.R. Shayan, Chair
Department of Electrical and Computer Engineering

_____ 20___ _____
Dr. Amir Asif, Dean,
Faculty of Engineering and Computer Science

Abstract

Single-channel speech enhancement based on deep neural networks

Zhiheng Ouyang

Speech enhancement (SE) aims to improve the speech quality of the degraded speech. Recently, researchers have resorted to deep-learning as a primary tool for speech enhancement, which often features deterministic models adopting supervised training. Typically, a neural network is trained as a mapping function to convert some features of noisy speech to certain targets that can be used to reconstruct clean speech. These methods of speech enhancement using neural networks have been focused on the estimation of spectral magnitude of clean speech considering that estimating spectral phase with neural networks is difficult due to the wrapping effect.

As an alternative, complex spectrum estimation implicitly resolves the phase estimation problem and has been proven to outperform spectral magnitude estimation. In the first contribution of this thesis, a fully convolutional neural network (FCN) is proposed for complex spectrogram estimation. Stacked frequency-dilated convolution is employed to obtain an exponential growth of the receptive field in frequency domain. The proposed network also features an efficient implementation that requires much fewer parameters as compared with conventional deep neural network (DNN) and convolutional neural network (CNN) while still yielding a comparable performance.

Consider that speech enhancement is only useful in noisy conditions, yet conventional SE methods often do not adapt to different noisy conditions. In the second contribution, we proposed a model that provides an automatic "on/off" switch for

speech enhancement. It is capable of scaling its computational complexity under different signal-to-noise ratio (SNR) levels by detecting clean or near-clean speech which requires no processing. By adopting information maximizing generative adversarial network (InfoGAN) in a deterministic, supervised manner, we incorporate the functionality of SNR-indicator into the model that adds little additional cost to the system.

We evaluate the proposed SE methods with two objectives: speech intelligibility and application to automatic speech recognition (ASR). Experimental results have shown that the CNN-based model is applicable for both objectives while the InfoGAN-based model is more useful in terms of speech intelligibility. The experiments also show that SE for ASR may be more challenging than improving the speech intelligibility, where a series of factors, including training dataset and neural network models, would impact the ASR performance.

Related Publications

The work in this thesis led to the following research publications:

1. **Zhiheng Ouyang**, Hongjiang Yu, Wei-Ping Zhu, Benoit Champagne. A Fully Convolutional Neural Network for Complex Spectrogram Processing in Speech Enhancement. *In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.* [Chapter 2]

2. **Zhiheng Ouyang**, Wei-Ping Zhu, Benoit Champagne. Speech Enhancement with Information Maximizing Generative Adversarial Network. *In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.*, Submitted October 2019. [Chapter 3]

3. **Zhiheng Ouyang**, Hongjiang Yu, Wei-Ping Zhu, Benoit Champagne. A Deep Neural Network Based Harmonic Noise Model for Speech Enhancement. *In Interspeech. 2018.*

Contents

List of Figures

List of Tables

List of Abbreviations

ASR Automatic Speech Recognition.

CIRM Complex Ideal Ratio Mask.

CNN Convolutional Neural Network.

CUDA Compute Unified Device Architecture.

CV Computer Vision.

DFT Discrete Fourier Transform.

DNN Deep Neural Network.

FAR False Alarm Rate.

FCN Fully Convolutional Network.

FRR False Reject Rate.

GAN Generative Adversarial Network.

GPU Graphics Processing Unit.

IBM Ideal Binary Mask.

IRM Ideal Ratio Mask.

iSTFT inverse Short-Time Fourier Transform.

KWS Keyword Spotting.

LPS Log-Power Spectrogram.

LSTM Long Short-Term Memory.

NLP Natural Language Processing.

NMF Non-negative Matrix Factorization.

PESQ Perceptual Evaluation of Speech Quality.

ReLU Rectified Linear Unit.

RIR Room Impulse Response.

RNN Recurrent Neural Network.

SDR Signal-to-Distortion Ratio.

SE Speech Enhancement.

SSNR Segmental Signal-to-Noise Ratio.

STFT Short-Time Fourier Transform.

STOI Short-Time Objective Intelligibility.

TPU Tensor Processing Unit.

VAD Voice Activity Detectiong.

VAE Variational Auto-Encoder.

WER Word Error Rate.

Chapter 1

Introduction: Deep Learning for Speech Enhancement

Enabled by recent advances in parallel computing in both hardware aspects, including graphics processing units (GPUs) and tensor processing units (TPUs), and software aspects including computing platforms such as compute unified device architecture (CUDA) and machine learning frameworks and libraries like tensorflow and pytorch, deep learning has firstly gained huge traction in the field of computer vision (CV) and then been introduced to numerous fields including natural language processing (NLP), audio and speech processing, and image processing.

It is believed that, traditional methods of speech processing which are often based on mathematical or statistical models and are somehow unintuitive, can be outperformed by deep learning methods on a large scale. Thus neural networks, including feedforward neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs) have been intensively studied for various tasks in speech processing such as speech synthesis, speech separation and speech enhancement. A majority of deep learning methods in speech processing are adopted from the field of CV. While

images which are analyzed in CV are of two dimensions by its nature (or three dimensions taking into account RGB channels), single-channel speech signals are one dimensional time-domain sequences with strong temporal correlations between samples. Interestingly, short-time Fourier transform (STFT) is often applied on the one-dimensional speech signal to obtain an image-like spectrogram, whose horizontal and vertical axes represent time and frequency, respectively. Yet there are still significant differences between the application context of the two fields that requires judicious modeling and adaption of the specific tasks in question.

This thesis investigates the deep-learning based speech enhancement methods, and specifically address a number of key factors that play a very important part in the algorithm design, including neural network models, objective functions, training dataset and strategy, and performance evaluation. Although some of these aspects are more empirical and sometimes lacks theoretical support, they are crucial for the general methodology of deep-learning based speech enhancement. The following briefly introduces the constituent components of those methods.

1.1 Neural network models for SE

In speech enhancement, neural network models often serve as a mapping function from some features, either carefully-engineered or raw, to targets such as clean speech itself or something that can be used to reconstruct clean speech (e.g., magnitude spectrogram of clean speech). Hence, for researchers in speech enhancement, deciding an appropriate neural network and choosing an appropriate feature-target set for the network are the first two problems to address. Treating them as separate tasks is a common practice when simple networks with fixed structures are adopted, such as fully-connected DNN and LSTM. One possible drawback of this approach is that the chosen feature-target set may not be optimal for specific neural network models,

thus may not produce a good performance in terms of both intelligibility and model complexity. Meanwhile, CNNs can be thought of as performing feature extraction and feature-target mapping jointly, which consequently, requires no prior feature selection, and could yield a joint optimization over feature extraction and mapping.

1.1.1 Input feature and output target

Various target sets, including mask-based targets [57, 58], spectrogram-based targets and time-domain waveform targets [8, 35, 41, 59], have been investigated and adopted to fit different network architectures.

Masks for speech enhancement consist of time-frequency elements that apply attenuation on the spectrogram of noisy speech to filter out the noise component:

$$Y(k,l) = X(k,l) \times M(k,l) \tag{1}$$

where $M(k,l)$ represents the mask, and $Y(k,l)$ and $X(k,l)$ denote the spectrogram of estimated clean speech and that of the noisy speech, respectively. Here, k is the time index and l is the frequency index.

As implied by its name, ideal binary mask (IBM) is a binary mask that roughly separates speech component from its noisy mixture:

$$IBM(k,l) = \begin{cases} 1, & \text{if SNR}(k,l) > LC \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $SNR(k,l)$ denotes the local SNR at time-index k and frequency-index l.

Compared to IBM, ideal ratio mask (IRM) is a more smooth mask where the value of the mask is the ratio of the spectrogram of clean speech to that of noisy speech, or a function of that:

$$IRM(k,l) = \left(\frac{Y(k,l)}{X(k,l)+1} \right)^{\beta} \tag{3}$$

where β is a tunable parameter that scales the mask and is usually set to 1.

Conventional speech enhancement methods such as DNN-based IRM only focus on enhancing the spectral magnitude and neglect the spectral phase of noisy speech, due to the lack of structure for spectral phase of speech [23]. As a walk-around solution to address this phase processing problem, complex ideal ratio mask (CIRM) is simply defined as an extension of IRM in complex-frequency domain:

$$CIRM(k,l) = \left(\frac{Y_c(k,l)}{X_c(k,l)}\right)^{\beta} \tag{4}$$

where $Y_c(k,l) = Y_r(k,l) + jY_i(k,l)$, i.e., $Y_c(k,l)$ is the complex spectrum of clean speech that consists of both real component $Y_r(k,l)$ and imaginary component $Y_i(k,l)$. Similarly, $X_c(k,l)$ is the complex spectrum of noisy speech. It has been shown that processing the complex spectrum is equivalent to processing spectral magnitude and phase simultaneously [8].

Given the fact that eventually the spectrogram of clean speech is obtained after applying the mask to the spectrogram of noisy speech, it is more straightforward to directly estimate the spectrogram of clean speech by neural network. As an alternative to spectrogram, raw waveform can also be an output target. However, waveforms are of high-dimensions and thus require some specific design of network structure to capture the high-level information.

While some studies [48, 55] suggest that masks may be a better target than spectrograms based on experimental results, it is noteworthy that the experiments in the studies were only conducted on a simple fully-connected DNN with a fixed feature set, thus it is very likely that the conclusion may not apply to most scenarios wherever there is a change in the experimental setup, e.g., network architecture, feature set, number of parameters, etc.

On the other hand, input feature set may have great impact on the performance of neural networks, especially on fully-connected DNN and LSTM. The authors of [55] have investigated the performance of a number of features, which are previously employed for robust ASR, on a fully-connected DNN for SE task. The features could

be arranged into several categories:

- Mel-domain features: mel-frequency cepstral coefficient (MFCC), delta-spectral cepstral coefficient (DSCC), log mel-spectrum feature (LOG-MEL)
- Linear prediction features: perceptual linear prediction (PLP), relative spectral transform PLP (RASTA-PLP)
- Gammatone-domain features: gammatone feature (GF), gammatone frequency cepstral coefficient (GFCC), gammatone frequency modulation coefficient (GFMC)
- Zero-crossing feature: zero-crossings with peak-amplitudes (ZCPA)
- Autocorrelation features: relative autocorrelation sequence MFCC (RAS-MFCC), autocorrelation sequence MFCC (AC-MFCC), phase autocorrelation MFCC (PAC-MFCC)
- Medium-time filtering features: power normalized cepstral coefficients (PNCC), suppression of slowly-varying components and the falling edge of the power envelope (SSF)
- Modulation domain features: Gabor filterbank (GFB), amplitude modulation spectrogram features (AMS)
- Pitch-based feature: Time-frequency features based on pitch tracking (PITCH)
- Multi-resolution feature: Multi-Resolution Cochleagram (MRCG)
- Spectral magnitude feature: log spectral magnitude (LOG-MAG)
- Time-domain feature: raw waveform (WAV)

Those features could be roughly divided into two groups: less-engineered plain features like log-magnitude spectrogram, and highly-engineered modulated features such

as multi-resolution cochleagram. Although Table 1 showed that Gammatone-based features are preferred over the others in terms of short-time objective intelligibility (STOI), it should be noted that the conclusion is only considered accurate on fully-connected DNN for IRM estimation at a certain range of SNR levels, and is probably not applicable whenever there is a change in network architecture, output target, or any different testing conditions. It is also worth mentioning that, using less-engineered features like log-magnitude spectrogram performs reasonably well while requires much fewer computations compared to highly-engineered features, and it is also relatively easy to take advantage of parallel computing, leading to a fast implementation. In practice, plain features such as spectral magnitude have also gained interest among many researchers because they are easy to obtain and perform reasonably well.

Overall, the choice of feature-target set has a great variety, and usually depends on the network architectures. Fully-connected DNN was once the most popular option for many researchers in SE, and thus highly-engineered feature-target set was therefore preferred as it can be used to train the network effectively and usually performs better. However, thanks to recent progress on CNN in the field of CV, it has been shown that CNN with certain structures can work considerably well on plain feature-target set such as magnitude spectrogram or complex spectrogram.

1.1.2 Neural networks

Deterministic models

Fully-connected DNN is the most basic and probably the most popular model adopted in SE. Typical use of fully-connected DNN for SE includes log-power spectrum estimation [59], IBM and IRM estimation [57], and CIRM estimation [58], etc. It consists of an input layer, hidden layers and an output layer. Rectified linear unit (ReLU) is most often employed among hidden layers to address the vanishing gradient problem, which refers to the observation that the absolute value of gradient becomes smaller

Table 1: STOI improvements (in %) for a list of features averaged on a set of test noises. *Sim. RIR* represents simulated room impulse responses and *Rec. RIRs* represent the recorded ones [55]

Feature	Matched noise			Unmatched noise		
	Anechoic	Sim. RIRs	Rec. RIRs	Anechoic	Sim. RIRs	Rec. RIRs
MRCG	**7.12**	**14.25**	**12.15**	**7.00**	7.28	8.99
GF	6.19	13.10	11.37	6.71	7.87	8.24
GFCC	5.33	12.56	10.99	6.32	6.92	7.01
LOG-MEL	5.14	12.07	10.28	6.00	6.98	7.52
LOG-MAG	4.86	12.13	9.69	5.75	6.64	7.19
GFB	4.99	12.47	11.51	6.22	7.01	7.86
PNCC	1.74	8.88	10.76	2.18	**8.68**	**10.52**
MFCC	4.49	11.03	9.69	5.36	5.96	6.26
RAS-MFCC	2.61	10.47	9.56	3.08	6.74	7.37
AC-MFCC	2.89	9.63	8.89	3.31	5.61	5.91
PLP	3.71	10.36	9.10	4.39	5.03	5.81
SSF-II	3.41	8.57	8.68	4.18	5.45	6.00
SSF-I	3.31	8.35	8.53	4.09	5.17	5.77
RASTA-PLP	1.79	7.27	8.56	1.97	6.62	7.92
PITCH	2.35	4.62	4.79	3.36	3.36	4.61
GFMC	-0.68	7.05	5.00	-0.54	4.44	4.16
WAV	0.94	2.32	2.68	0.02	0.99	1.63
AMS	0.31	0.30	-1.38	0.19	-2.99	-3.40
PAC-MFCC	0.00	-0.33	-0.82	0.18	-0.92	-0.67

progressively during back-propagation, thus resulting in a poor training as the parameters of lower layers being not modified effectively. The structure of the output layer, in particular, can be modified in many ways if the output is a combination of different targets. For example, Fig. 1 shows a fully-connected DNN for CIRM estimation, whose output layer consists of 2 sub-networks which use their own local connections to produce spectral masks for real and imaginary components, respectively.

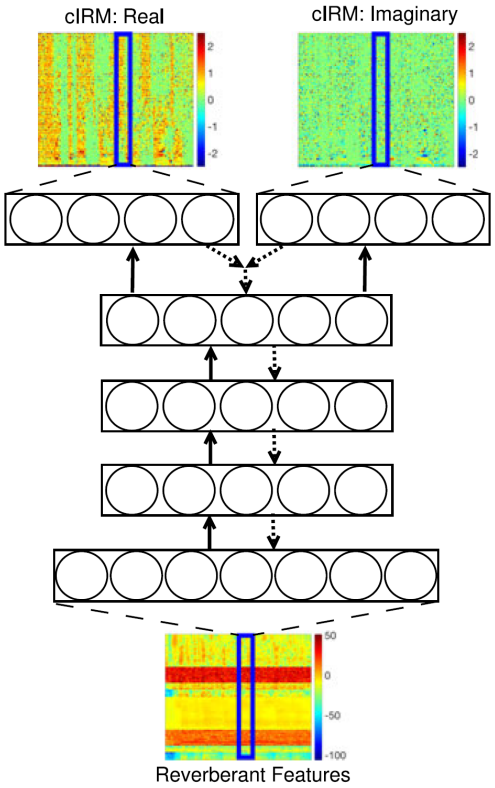
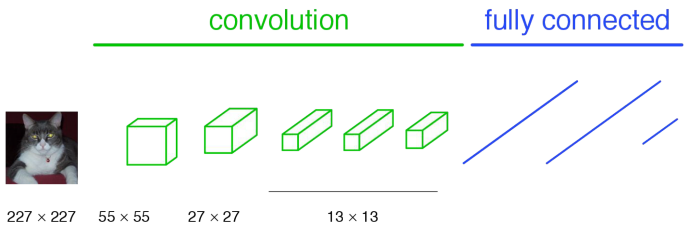


Figure 1: A DNN for CIRM estimation [58]

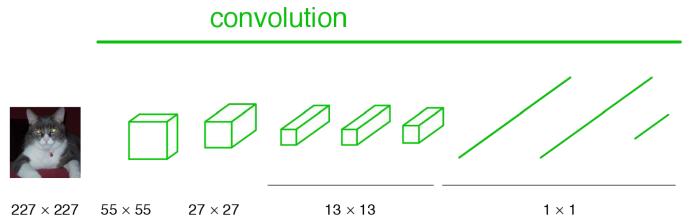
Unlike fully-connected DNN, the structure of CNN is highly customizable which results in a great variety of design. A number of CNN structures have been explored for SE tasks, including SE-WaveNet [41], dilated residual CNN for mask estimation [50], and redundant convolutional auto-encoder [35], etc. The classic structure of CNN takes 2D images (3D if channel counts as one dimension) as input and performs

feature extraction by stacking pairs of a convolutional layer and a pooling (down-sampling) layer, and appending a couple of fully-convolutional layers at the end to obtain the output of any desired shape. The CNN of this structure could be simply employed in frequency-domain as spectrogram are of 2 dimensions just like images. In [8], a CNN with a similar structure is employed for SE task via complex spectrogram estimation. Yet in many recent works, the pooling layers are often discarded or replaced by using stride and dilation with convolution, because it is argued that the pooling operation may be too destructive if the objective is to reduce the spatial size. Furthermore, researchers have proposed to replace fully-connected layers in the classic structure by 1d convolutional layers to obtain a fully convolutional network (FCN). Long et al. demonstrated that FCN can efficiently learn to perform per-pixel-prediction tasks such as semantic segmentation. Park and Lee proposed a FCN with autoencoder-decoder architecture for SE task via magnitude spectrogram-mapping and demonstrated that a CNN with a proper network structure could yield a comparable SE performance as fully-connected DNN and RNN while being significantly more compact than both networks, as illustrated in Fig. 3.

Rethage et al. employed WaveNet [51] for SE task, which features a heavy use of dilated 1d convolution with skip-connection and residual learning. As is shown in Fig. 4, the network is fully convolutional, and in particular, the filters are all designed to be 1-dimensional. While 1d convolution is an efficient way to process time-domain speech, adopting plain 1d convolution may not be ideal as the network would fail to be trained effectively due to its limited receptive field. Specifically, the time-domain speech consists of dense and highly-correlated data due to the high sampling rate (e.g., the number of samples reaches $16k$ per second with a sampling rate $16k$ Hz), yet stacking regular 1d convolution simply leads to a linear growth of the receptive field, which makes it impossible for CNN to capture the contextual information and perform feature extraction. The problem can be addressed by incorporating dilation within



(a) a CNN with fully-connected layers



(b) a fully convolutional neural network

Figure 2: Replacing fully-connected layers with 1×1 convolutional layers for semantic segmentation task [54]. The numbers denote the spatial size of feature maps in both (a) and (b).

cascading convolutional layers, which allows the receptive field to grow exponentially. We will give a more detailed introduction about CNN of this type in Chapter 2.

RNN, especially LSTM, has been studied for speech enhancement task for its ability of modeling temporal relations. The use of RNN is similar to that of fully-connected DNN in SE, thus a fully-connected DNN could be simply replaced by a RNN without major modifications. While fully-connected DNNs are mostly designed to work in frequency-domain and the input usually contains a number of frames to consider the contextual information between the previous and current data, RNN, in contrast, only requires the single-frame input because it can model the contextual relation between two consecutive frames of the input. As an example, Fig. 5 shows a LSTM for CIRM estimation, where the only difference between this network and that in [58] is that the fully-connected DNN is replaced by LSTM and the input only consists of one frame. It is noteworthy that, while it is reasonable to use fewer units among the hidden layers of LSTM than that of fully-connected DNN, this does not mean the size of the model for LSTM is anywhere smaller than that of fully-connected

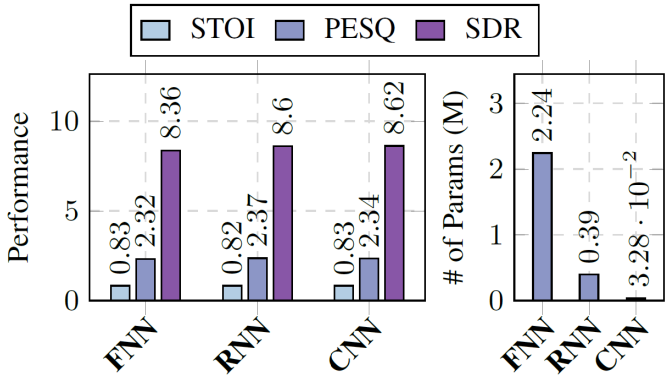


Figure 3: A comparison of SE performance of different network structures and their model size [35]. The FNN denotes fully-connected DNN.

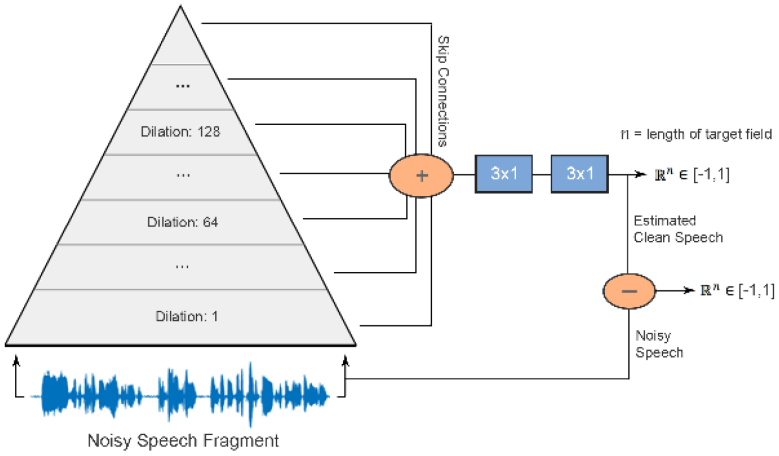


Figure 4: Wavenet for SE

DNN since LSTM uses multiple sets of weights and biases to model the contextual information while the fully-connected DNN only uses one.

Generative models

While the deterministic models described above have been proven suitable for SE and are more straightforward than generative models in the context of SE task, generative models such as variational auto-encoder (VAE) and generative adversarial network (GAN) have also been investigated. It is claimed that, even though the deterministic model for SE adopts a supervised approach which already requires a fairly large

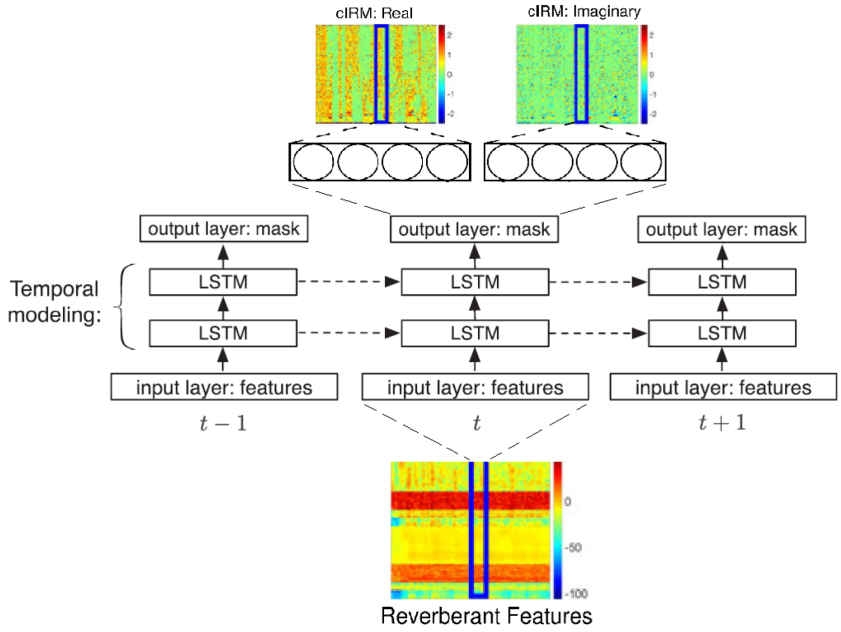


Figure 5: LSTM for CIRM estimation

amount of training data, it is still not very robust against unknown environments as it is not always possible to cover all noisy scenarios in real-world applications during training. [4].

A generative model usually does not make assumptions on the environmental noise, because it is often trained to generate clean speech, which means unsupervised training is employed and the training data only contains clean speech. For example, in [4], a VAE is employed as a generative model that is capable of generating the spectrogram of clean speech. In the context of SE, the model is adapted in a statistical approach to obtain clean speech from the noisy one. As shown in Fig. 6, the noisy speech is used as observed data to create a posterior distribution which the spectrogram of clean speech can be sampled from. The posterior distribution aims to minimize the semantic loss between the generated clean speech and the noisy speech.

GAN, as a generative model, could be employed in a similar manner. However, it should be mentioned that, while GAN has been studied for SE tasks in a number of works, it is always adopted in a deterministic manner with supervised training, i.e.,

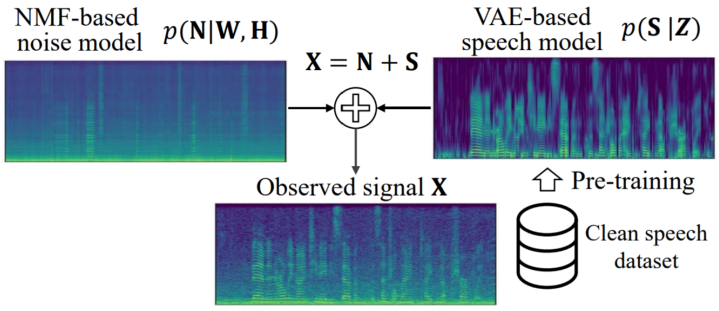


Figure 6: The VAE-based SE method [4]. X, N and S refer to the spectrograms of noisy speech, noise and clean speech, respectively, Z represents the latent variable sampled from the standard Gaussian prior, W denotes the spectral basis vectors and H denotes the activation vectors, both of which are used to represent the spectrogram of noise.

the GAN is employed for explicit spectrum mapping, which is essentially the same as a deterministic model. We suggest that a similar approach as [60], which is designed for image inpainting in the field of CV, be adopted in the context of SE. Illustrated in Fig. 7, the input latent vector of GAN is updated iteratively to minimize the contextual loss between the masked image and the output image of GAN. To adapt the method, the spectrogram of noisy speech could be transformed into a masked spectrogram by energy-thresholding, and a specific loss function of noisy and clean spectrograms should be designed and minimized to find the best match to the spectrogram of the clean speech given the masked noisy one.

1.2 Training datasets and training strategy

The bias-variance tradeoff [10] is a long standing problem in machine learning which has to be considered when deploying deep-learning models for real-world applications. It could be illustrated by a typical observation that, the better the model fits the training data, the less likely that it will generalize well on unseen data, or vise versa. Specifically, a model trained with high-variance learning method would be able to represent the training data well, which unfortunately may lead to overfitting and

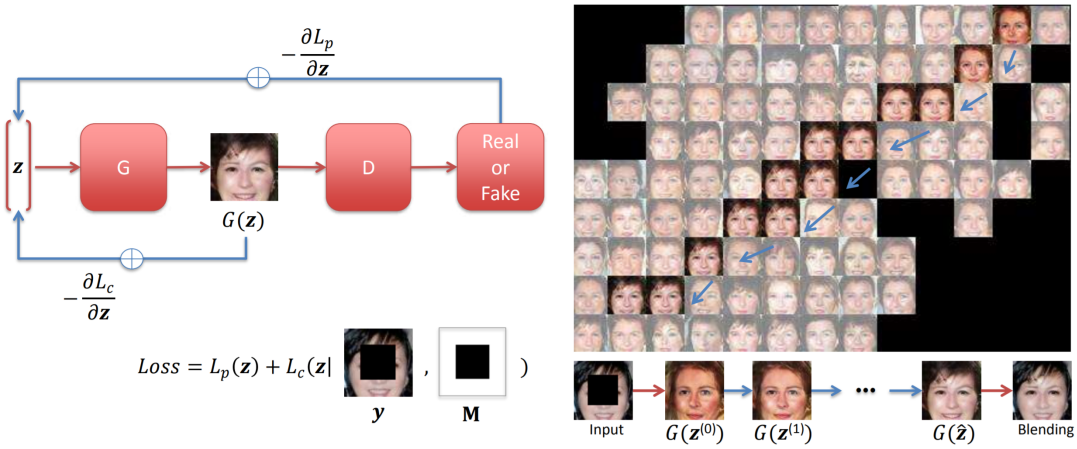


Figure 7: A GAN for image inpainting designed by a generative approach [60]. **y** is the masked input image and **M** denotes the mask. The output is the corresponding inpainted image. **Left**: The latent vector **z** is updated iteratively based on a designed loss function to find the closest mapping to the masked image. **Right**: The output image manifold changes as **z** gets updated.

could produce poor results on unpresented data. In contrast, although the model trained with high bias does not overfit the training data, which consequently yields a simpler model, it may underfit the presented data and fails to capture some important regularities.

This bias-variance problem causes a series of problems when applying deep learning models for SE. Typically, it makes the evaluation of different algorithms rather unreliable, because the results often fluctuate depending on the training and testing data as well as the metrics adopted. A common practice is to compare the model with some other methods of its kind with a specific dataset, yet the comparison is questionable even if all methods are trained on the same dataset, because it is usually uncertain if the testing data is in favor of high-variance model or high-bias model. For example, a high-variance model may easily outperform high-bias models on testing data that is similar to training set, yet outperformed by same high-bias models on testing data that is relatively unseen in training dataset.

Another example of the bias-variance tradeoff in SE is when the algorithms are

evaluated with metrics covering different aspects. For example, high-biased models may be preferable to high-variance methods for ASR metrics, while the later may be more useful for general intelligibility metrics such as PESQ as indicated in [6].

While using a large and diverse training dataset does not directly resolve the dilemma, it is still considered very helpful for the evaluation of a model, and is particularly vital for deploying a model for real-world applications. The benefits of using a large dataset for training is straightforward: for high-variance models, it provides more instances, which hopefully covers more unseen cases where the high-variance model fails to work, and for high-bias models, it simply helps ease the training process as some important regularities may become obvious as more data is provided. The training dataset for SE usually consists of two parts: speech dataset and noise dataset.

1.2.1 Speech dataset

While a number of speech datasets could be used for SE tasks, in particular, the ones originally designed for ASR are preferred over the others because of their size and diversity. The characteristics concerning the speech dataset could be divided into two aspects: audio characteristics and text characteristics. Audio characteristics include the total duration of speech, the number of speakers and the distribution of speakers' gender and age group, speech quality, dialect and accent, tempos of speech, etc. Text characteristics refer to the language of speech and the richness of language such as the number of words, etc. For SE task, important characteristics include the duration of speech, the number of speakers and the distribution of speakers' gender, speech quality and the language of speech. Below, we briefly introduce some speech datasets that are considered as suitable for training SE models:

1) TIMIT [9]: TIMIT is often used in SE tasks for both training and evaluation purposes. It consists of 6300 utterances from 192 female speakers and 438 male

speakers, each speaking 10 short annotated sentences. With a total duration of approximately 5 hours, it also features 8 major dialect divisions of American English.

2) TED-LIUM [18, 44, 45]: TED-LIUM is a public domain speech dataset that contains hundreds of hours of TED [2] talks in English language. The first version [44] consists of a total duration of 188 hours of talks, which come from 666 speakers (about 70 percent of the audio are from male speakers) with a mean duration of 9 minutes per talk. The second [45] and third versions [18] continue to enrich the dataset as more TED talks are added. Designed for the training of ASR systems, it is fairly rich in terms of languages, with $2.56m$ words covered in talks. The audio is accompanied with automatically aligned transcripts, which is also useful for identifying the speech segments. As the talks were previously recorded in conference halls that were full of audiences, the audio is noticeably reverberant and noisy (with the appearance of sound such as cough and applause).

3) LibriSpeech [34]: LibriSpeech is an archive of public domain English audiobooks from LibriVox [1]. It contains about $1k$ hours of speech that is sampled at 16 kHz. The speech comes from approximately $2.5k$ speakers, and about 52 percent of the speakers are male. The mean duration is limited to 25 minutes to avoid major imbalances in per-speaker audio duration.

4) MUSAN [47]: As its name suggests, music, speech and noise corpus (MUSAN) is a complementary dataset that comprises audios of music, speech and noise. As described in the original paper, this dataset is suitable for training models for voice activity detection (VAD) and music/speech discrimination. The speech portion of the dataset contains about 60 hours of speech, with about 20 hours of multi-language speech from LibriVox and 40 hours of collected recordings from

US government (federal and states) hearings, committees and debates. About half of the LibriVox audios are in English and the rest is in 11 other languages. The recordings of US government hearings are completely in English.

1.2.2 Noise dataset

Noise dataset is a collection of recordings of different soundscapes. Ideally, it should contain a wide range of sound types to minimize the gap between training and real-world applications. While there exist certain datasets initially designed for training and evaluating SE models, many of them only provide recordings of non-stationary noise whose types are rather limited. Another major drawback is that plenty of them are not in public domain and thus often require licensing, making it hard for researchers to obtain. For example, a commonly-used dataset in SE is NOISEX-92 [53], which provides 5 types of noise: voice noise (babble), factory noise, HF radio channel noise such as pink noise and white noise, military noises including fighter jets (Buccaneer, F16), destroyer noises (engine room, operations room), tank noise (Leopard, M109) and machine gun, and lastly, car noise (Volvo). Though it may be suitable for the initial evaluation, the dataset lacks a variety of soundscapes in daily life, thus it does not meet the demand of most real-world scenarios.

Realizing the fact that, depending on the context, noise could literally be any kind of sound, a more convenient way is to utilize datasets that are originally designed for tasks such as sound event detection where a wide range of sound is presented. As an example, ESC-50 dataset [37] contains 50 types of labeled recordings which could be loosely divided into 5 major categorizes that are shown in Table 2. The dataset has a total duration of around 2 hours and the mean duration of each type of sound is approximately 2.4 minutes.

Table 2: The types of sound in ESC-50 dataset

Animals	**Natural soundscapes & water sounds**	**Human, non-speech sounds**	**Interior, domestic sounds**	**Exterior, urban noises**
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

1.2.3 Training strategy

The training strategy generally depends on the specific deep-learning models. For SE task, it is often straightforward to adopt supervised training as the task can be categorized into sequence regression [39]. A deterministic neural network model is designed as a unique mapping function from features to certain target and is trained with a loss function that is usually defined using the mean-square distance or $\ell 1$-norm between the true value of the target and its prediction. Most existing methods are designed in this deterministic manner and usually yield a promising performance. The training targets are sometimes preprocessed and transformed, e.g., the DNN-based CIRM adopts hyperbolic tangent transform to compress its range of values from $(-\infty, \infty)$ to $(-1, 1)$.

Though not being the mainstream, unsupervised, or semi-supervised training could also be applied on certain generative models such as variational auto-encoder (VAE) and generative adversarial network (GAN). The model is often trained on clean speech dataset in order to model the speech with its probability distribution. For example, in [4], a VAE is adopted along with non-negative matrix factorization

(NMF) and is trained on clean speech in advance and then adapted to noise on the fly. It is claimed to outperform conventional DNN-based methods in unseen noisy environments.

While there are barely any so-called best practices for training the neural networks for SE tasks because the training process varies on different models and usually requires a lot of experiments and tweaking of the network and related parameters, a few practices are commonly considered appropriate in order to make the neural network models suitable for real-world applications. For example, the speech data used for training should be gender-balanced covering all age groups, and the duration of speech per speaker should be unified. The noise data used in training should be representative and cover common scenarios like urban soundscapes. On the other hand, *RMSprop* [13] and *Adam* [21] optimizers are often applicable for training neural network models for SE task. *RMSprop* uses the average of the magnitude of the recent gradients while *Adam* considers the moments of the gradients. Both optimizers provide a more stable training over classic stochastic gradient descent (SGD).

1.3 SE Objective and evaluation

While the overall task of speech enhancement is to improve the speech quality in noisy speech environments, specific application scenarios such as mobile communication, hearing aids and automatic speech recognition (ASR) may have different concerns on the objectives and performance metrics. For example, for hearing air and mobile communication, improving the intelligibility of speech is a main goal, while on ASR the main focus is to reduce the word error rate (WER) to improve the noise-robustness of the system.

It should be noted that the performance of a SE system on certain objectives and metrics does not necessarily relate to the performance on the others. For example,

even if a SE system yields improvement on intelligibility, WER is not necessarily guaranteed to be improved. Empirically optimization for WER is a more challenging task than improving intelligibility as the ASR system sometimes serves as a black box, yet it is believed that a joint optimization between ASR and SE system is required in order to avoid the sub-optimization problem that could lead to an unstable performance of the system.

1.3.1 Evaluation of intelligibility

Most commonly-used metrics for the evaluation of speech intelligibility could be divided into perception-level and signal-level [55]. Perception-level metrics include perceptual evaluation of speech quality (PESQ) [42] and short-time objective intelligibility (STOI) [20, 49]. PESQ is a standard metric which predicts the perceptual mean opinion score (MOS) of a given speech in a interval of -0.5 to 4.5. The auditory transform is applied on both reference clean speech and the speech to be measured to obtain their corresponding loudness spectrograms. MOS is then predicted by comparing the two loudness spectrograms. On the other hand, STOI reflects the short-time intelligibility in a range of 0 to 1, and it is achieved by measuring the correlation between the short-time temporal envelopes of the reference speech and the speech to be evaluated. Although STOI has been reported to tend to over-predict the intelligibility scores [17, 24], it is still considered accurate as it consistently correlates with human intelligibility.

Signal-level metrics quantify the degree of distortion with respect to clean reference at the signal level. Popular metrics comprise signal-to-distortion ratio (SDR) and segmental signal to noise ratio (SSNR). It is noticeable that, while PESQ and STOI model subjective intelligibility test scores which directly reflect the level of intelligibility, the results on SDR and SSNR are usually consistent with PESQ and STOI, as the more clean the speech is, the more intelligible to human ears.

1.3.2 Evaluation of ASR performance

For the evaluation of ASR performance, the metrics are WER for large vocabulary continuous speech recognition (LVCSR) systems, and false reject rate (FRR) and false alarm rate (FAR) for keyword spotting (KWS) systems. It should be noted that, as multi-style training (MTR) has been employed in state-of-the-art ASR systems and effectively improves the noise robustness [6, 26], a SE system that serves as pre-processor may still be beneficial. However, it has been shown that adopting a SE model for this purpose requires retraining the MTR model with both noisy and enhanced features, otherwise it may degrade the ASR performance compared to the ASR without any enhancement [6].

1.4 Objective and organization of this thesis

1.4.1 Objective of the thesis

The main objective of this thesis is to develop single-channel speech enhancement methods with deep neural networks for noise reduction.

In the first approach, a fully convolutional network is proposed for complex spectrogram estimation in speech enhancement. The network adapts the WaveNet framework [51] for 2-dimensional spectrogram processing. Stacked frequency-dilated convolution is employed to obtain an exponential growth of the receptive field in frequency domain, which enables an efficient implementation that requires much fewer parameters as compared with conventional fully-connected DNN and CNN while still yielding a comparable performance.

Although speech enhancement is useful in noisy conditions, it is often redundant for clean speech. As most voice activity detection (VAD) techniques that are adopted as pre-processor in SE methods are not designed to identify the SNR levels, conventional SE methods usually do not adapt to different noisy conditions. In the second

contribution, we proposed a model that considers different SNR levels and provides an automatic "on/off" switch for speech enhancement. It is capable of scaling its computational complexity under different SNR levels by detecting clean or near-clean speech which requires no processing. By adopting information maximizing generative adversarial network (InfoGAN) in a deterministic, supervised manner, we incorporate the functionality of SNR-indicator into the model that adds little additional cost to the system.

1.4.2 Organization of the thesis

The rest of this thesis is organized as follows:

Chapter 2: This chapter first introduces complex spectrogram processing in speech enhancement and some network structures that are suitable for this task, including fully-connected DNNs and CNNs. It then gives detailed description of the proposed fully convolutional network for complex spectrogram estimation. Comparative study of the proposed method with other existing methods is also presented.

Chapter 3: This chapter describes the generative adversarial network and its application in speech enhancement. It introduces the information maximizing generative adversarial network (InfoGAN) and the proposed InfoGAN-based SE method. Experimental results of the proposed approach and related methods are given on the aspects of speech intelligibility and KWS.

Chapter 4: This chapter concludes this thesis and gives some directions for future research.

Chapter 2

Single-Channel Speech Enhancement with Convolutional Network

In this chapter, we develop a method for single-channel speech enhancement using convolutional neural network. This chapter is organized as follows. In this section, we introduce background and recent works in CNN-based speech enhancement methods. Section 2.2 describes the proposed CNN model for single-channel speech enhancement. Performance of the proposed CNN method is evaluated in Section 2.3.

2.1 Previous work

2.1.1 Complex spectrogram processing by DNN

Conventional deep-learning-based SE methods process speech signal in frequency domain, where the main focus is to obtain the spectral magnitude of clean speech. Given a noisy speech y, a common practice is to first apply short-time Fourier transform (STFT) to the signal: $Y = \text{STFT}(y)$, where Y refers to as the spectrogram and

is a matrix with size of $K \times L$, with K representing the number of time segments and L the number of frequency bins. While Y takes natural complex-valued, i.e., $Y = Y_r + Y_i$ where Y_r represents the real STFT component and Y_i the imaginary STFT component, the spectral magnitude is simply $|Y|$, the absolute value of Y, and phase is defined as $\phi_y = arctan(Y_i/Y_r)$. The spectral magnitude of clean speech, $|X|$, is then estimated as $|\hat{X}|$ with a neural network and transformed back to time domain along with the spectral phase of noisy speech by using inverse short-time Fourier transform (iSTFT): $\hat{x} = \text{iSTFT}[|\hat{X}|e^{j\phi_y}]$. A number of methods have been developed in this manner. To name a few, the DNN-based log-power spectrum estimation [59], the DNN-based ideal ratio mask [57], the CNN-based spectrogram estimation [35]. For example, the DNN-based log-power spectrum estimation [59] employs a fully-connected DNN to estimate the log-power spectrogram (LPS) of clean speech given that of noisy speech as shown in Fig. 8. The detailed block diagram of the corresponding SE system is shown in Fig. 9. Specifically, the time-domain noisy speech is first transformed into frequency domain by computing the discrete Fourier transform (DFT) of each overlapping windowed frames, the result of which is often called spectrogram. The logarithm of the squared value of the spectrogram is calculated to obtain the log-power spectrogram, which is then fed into the neural network as input. The output of neural network is transformed back into spectral magnitude of the estimated clean speech, which, along with the spectral phase of noisy speech, is used to synthesize the estimated clean speech in time domain.

It should be noted that, while there are some methods that directly process speech signal in time domain [36, 41] which yields an end-to-end solution, most of the existing algorithms are designed to work in frequency domain given the fact that it's relatively fast (time-domain convolution equals element-wise multiplication in frequency domain) and the harmonic structure of clean speech is disentangled from the mixture in frequency domain.

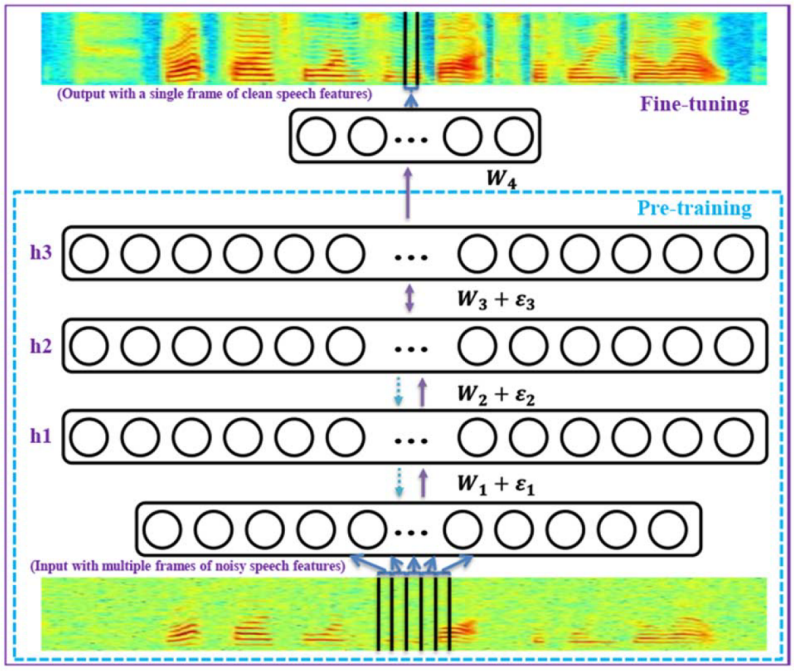


Figure 8: A DNN for LPS estimation. The DNN contains three hidden layers with 2048 units per hidden layer. The input of DNN consists of several consecutive frames of LPS of noisy speech and the output is a single frame of LPS of estimated clean speech [59].

As described, most of the spectral-based speech enhancement algorithms only modify the magnitude of the STFT components and leave the phase untouched, because it is generally considered that the magnitude carries most of the information of a signal. Yet recent studies have shown that employing spectral phase can further improve the perceptual quality of speech [11, 23, 33]. Specifically, Krawczyk and Gerkmann [23] showed that the perceptual evaluation of speech quality (PESQ) could be improved by around 0.2 when using the combination of noisy magnitude and estimated phase for speech reconstruction. While it may be beneficial to process noisy phase for a better denoising performance, it remains difficult to directly estimate the true phase of clean speech from noisy phase using deep learning, possibly due to the wrapping effect and the lack of phase structure in human speech [8, 58].

CNNs have been shown to be able to directly process speech signal in time-domain

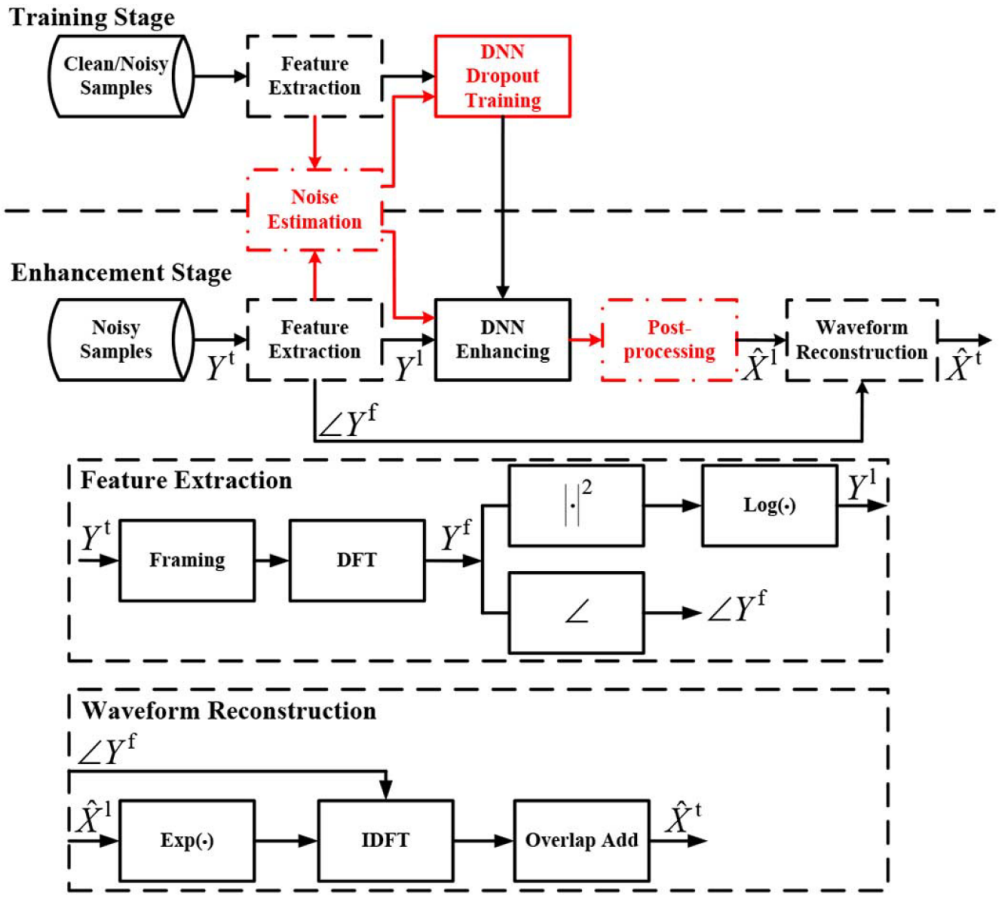


Figure 9: The block diagram of DNN for LPS estimation. $\mathbf{Y}$ denotes noisy speech and $\mathbf{X}$ denotes clean speech [59].

[36, 41], where the spectral magnitude and phase are jointly processed in an implicit way. However, models of this kind are often outperformed by frequency-domain-based methods, possibly because the structure of speech in time domain may be overwhelming for neural network to capture, which, on the other hand, is greatly disentangled in frequency domain in view of the harmonic structure of spectral magnitude of speech.

Another possible walk-around of phase processing is through complex spectrogram estimation. As shown in Fig. 10, Williamson et al. [58] found that the structure of complex spectrogram is similar to that of magnitude spectrogram, which makes it feasible to simply reuse a neural network that is originally designed for spectral magnitude estimation for estimation, while maintaining another neural network for

magnitude processing.

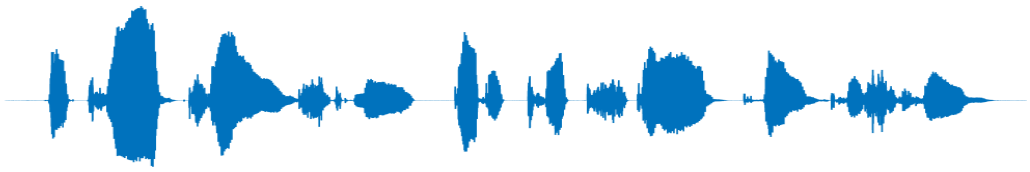
A few speech enhancement methods have been developed based on the estimation of complex noisy spectrogram, in which the noisy phase is implicitly processed. Williamson et al. [58] proposed a DNN-based masking technique to estimate a complex-valued mask from a spectral feature set by employing a fully-connected DNN, where the hidden layers of the fully-connected DNN are shared, and both real and imaginary components of the complex mask are simultaneously estimated by the output layers of the DNN. The resultant model is reported to improve the SE performance as compared with the fully-connected DNN employed for ideal ratio mask (IRM) estimation.

On the other hand, Fu et al. [8] employed a CNN to estimate the complex spectrogram of clean speech directly from the noisy one. As shown in Fig. 11, the structure of CNN consists of both convolutional layers and fully-connected layers. Though both methods are reported to have achieved a better denosing performance compared with DNN-based magnitude-processing methods, no further evidence is given to show the effectiveness of phase estimation through complex spectrogram processing.

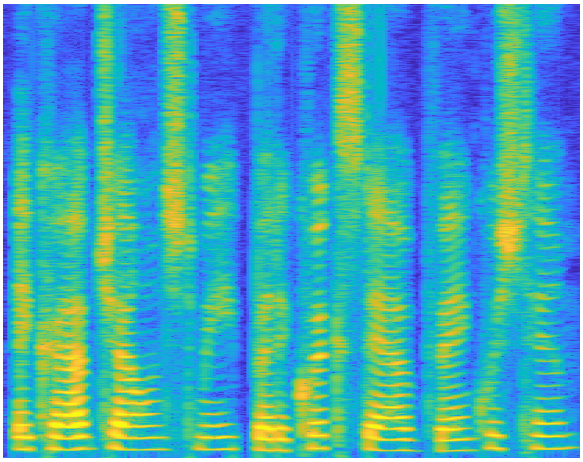
2.1.2 Convolutional neural networks for speech enhancement

Fully-connected deep neural network remains as the most common choice for speech enhancement tasks and is often adopted as a non-linear mapping function between noisy features and clean ones [32, 57–59]. Researchers have attempted to replace the fully-connected DNN by other neural network models, among which CNN has gained increasing popularity thanks to the recent progress in the filed of CV.

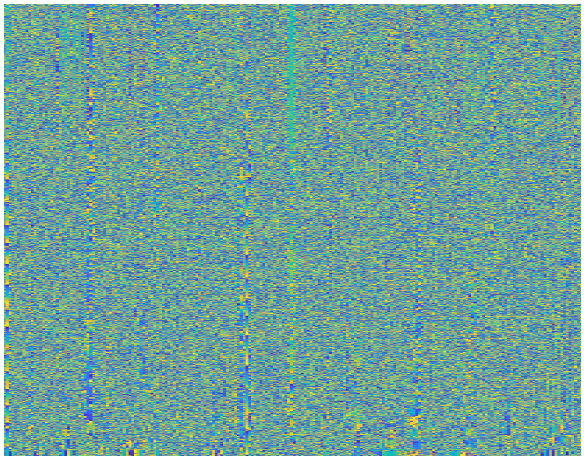
CNNs are known to be highly customizable and flexible in terms of the network design. Various network structures have been introduced and investigated for SE task. A classic CNN structure shown in Fig. 11 is employed in [8]. This conventional design



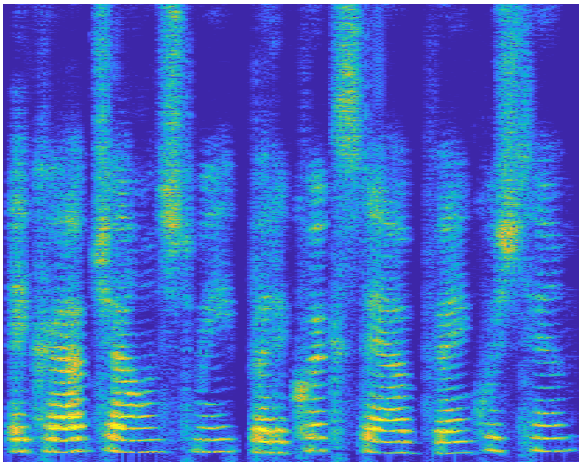
(a) A speech signal in time-domain



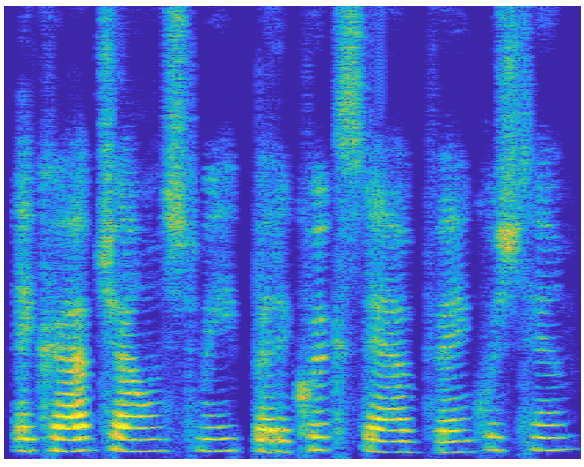
(b) Spectral magnitude



(c) Spectral phase



(d) Real STFT component



(e) Imaginary STFT component

Figure 10: Different representations of a clean speech signal

features a joint use of convolutional layers and fully-connected layers (sometimes called *dense layers*). Intuitively, convolutional layers could be considered as a feature extractor which produces some high-dimensional features. These features are then fed into the fully-connected layers that map the noisy features of speech to some targets that are latter used to reconstruct the clean speech.

Park and Lee [35] proposed a fully convolutional network (FCN) for speech denosing, where CNN is employed to extract the features for the reconstruction of the clean speech. The proposed CNN is a variant of the encoder-decoder network named Redundant Convolutional Encoder-Decoder network (R-CED) as shown in Fig. 12. The classic encoder-decoder design shown in Fig. 13 is a popular CNN structure that is often adopted as the structure of generator in generative adversarial network (GAN). Ideally, the encoder compresses the input into some high-dimensional features and the decoder reconstructs the data based on the high-level features. The spatial size of feature maps often shrinks (which refer to *downsampling*) among the encoder layers to compress the input to some latent representation, and then expands among the decoder layers (*upsampling*) to reconstruct the data, and meanwhile, the number of feature maps increases within the encoder and decreases within the decoder. As opposed to this, the R-CED uses a different way to produce the high-level representation and reconstruct data. Assuming the high-dimensional representation of the input can be provided by enough feature maps, downsampling is abandoned among the encoder layers, and it is replaced by increasing the number of channels of filters. Consequently, upsampling is replaced by decreasing the number of channels within the decoder layers.

It should be mentioned that, although both classic CNN with dense layers and fully convolutional network (FCN) could be considered as the non-linear mapping function in SE task, FCNs are often more restricted as they require the input feature and output target to have the same form (e.g., both to be magnitude spectrogram).

The conventional CNNs that consist of fully-connected layers, on the other hand, usually do not have this problem. The possible reason is that, the receptive field of fully-connected layers is generally larger than that of convolutional layers due to their difference in connectivity. The convolutional layers use a local connectivity that is limited by the size of the filters, while the fully-connected layers use a full connectivity (as suggested by its name), which, on the other hand, often leads to a greater number of parameters to model this connection.

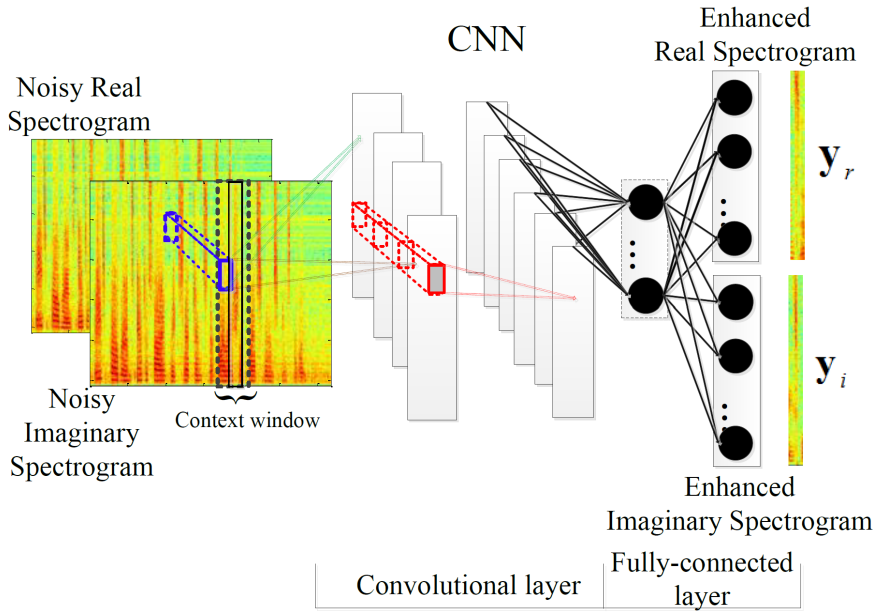


Figure 11: A CNN for complex spectrogram estimation [8]

2.1.3 WaveNet framework

WaveNet [51] is a generative convolutional model designed for audio synthesis and is proposed by DeepMind, Google. As shown in Fig. 14, dilated 1d convolution is stacked by many layers and the dilation is increased by a factor of 2, which allows for an exponential expansion of the receptive field. For example, as illustrated in Fig. 14, stacking 4 dilated 1d convolutional layers with filter size of 2 can produce a receptive field with a size of 16. In practice, the dilation factor is doubled for every layer up to

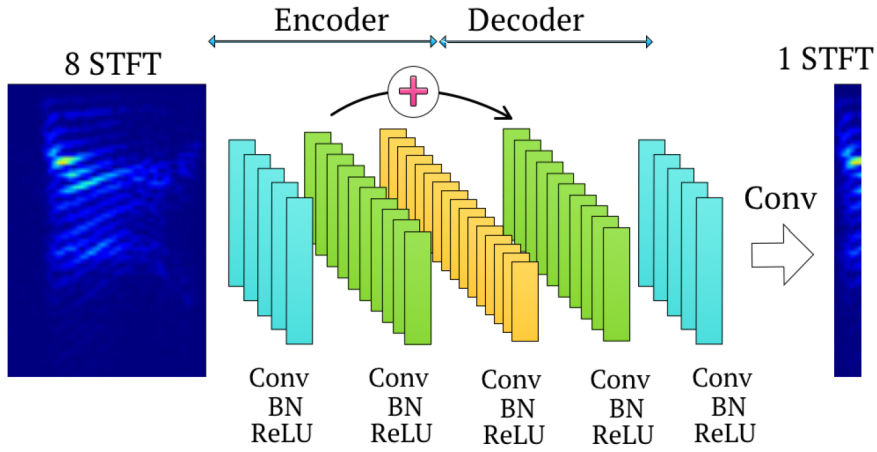


Figure 12: The network sturcture of redundant convolutional encoder-decoder (RCED) [35]

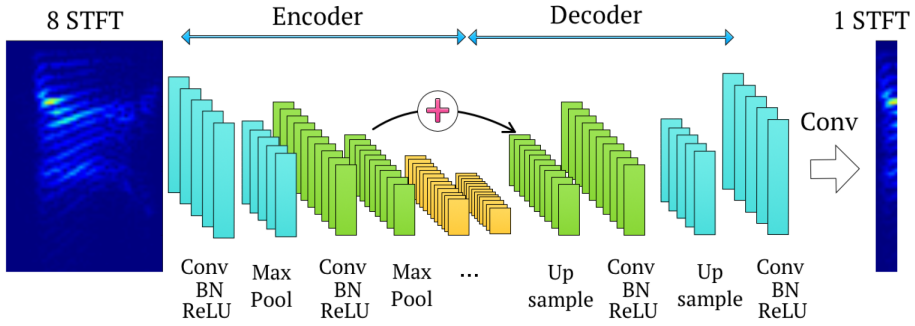


Figure 13: A FCN with autoencoder-decoder design [35]

512 and then repeated, e.g., 1, 2, 4, .., 256, 512, 1, 2, 4, .., 256, 512, 1, 2, 4, ..., 256, 512.

In order to fit the context of autoregressive audio generation, the CNN models the joint conditional probability of the waveform $\mathbf{x}$:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_1, ..., x_{t-1}) \tag{5}$$

where x_t represents the audio sample at time step t, which is conditioned on the samples at all previous time steps. Naturally, the convolution must obey the causality, as the model cannot violate the sequence of prediction: the audio sample x_t cannot depend on the future time steps, i.e., x_{t+1}, x_{t+2}..., which, in WaveNet, is called causal

convolution and implemented by shifting the output of a normal convolution in time domain.

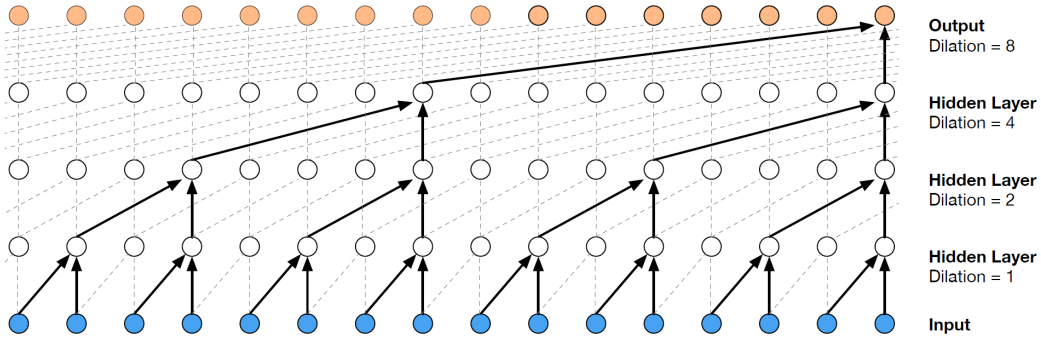


Figure 14: Stacked dilated causal 1d convolution [51]. Solid lines refer to the current dilated causal convolution and the dashed lines refer to the past and future convolution.

Fig. 15 further illustrates its network architecture, which features a gated activation function [52]:

$$\mathbf{z} = \tanh(\mathbf{W}_{f,k} * \mathbf{x}) \odot \sigma(\mathbf{W}_{g,k} * \mathbf{x}) \tag{6}$$

where $\mathbf{W}$ denotes the convolution filter, k indexes the layer, f represents filter and g represents gate, $*$ and $\odot$ represent convolution operation and element-wise multiplication, respectively, and $\sigma(\dot{})$ is the sigmoid function.

As shown in the figure, the network stacks a number of convolution layers which consists of the basic components: dilated convolution, gated activation function and 1×1 convolution. The residual learning and skip connection are used across different layers, which further take advantage of the efficient 1×1 convolution to perform the channel mapping.

The WaveNet has been successfully applied to SE [41]. As illustrated in Fig. 4, the SE WaveNet keeps the same processing block while employing the non-causal, dilated 1d convolution, which essentially change the functionality of the network from the sequential, non-parallelizable time-domain audio sample prediction to parallizable regression. As shown in Fig. 16, the input to SE WaveNet consists of previous samples

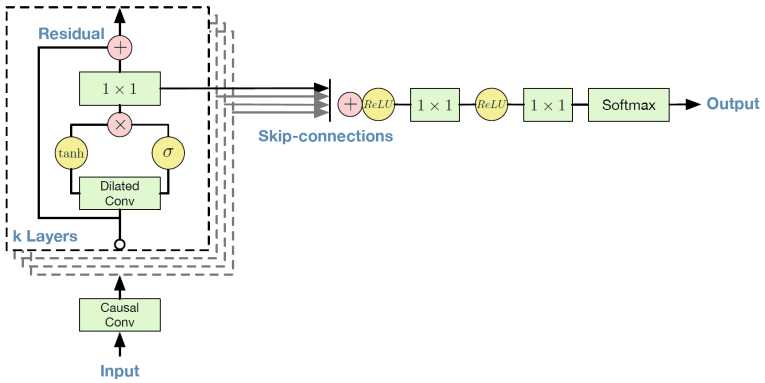


Figure 15: The network structure of WaveNet [51]

and future samples. While the SE task does not require causality, it introduces some latency as future samples are a part of the input. Though the result does not compare with modern deep-learning-based models, it is reported to outperform traditional Wiener filtering method.

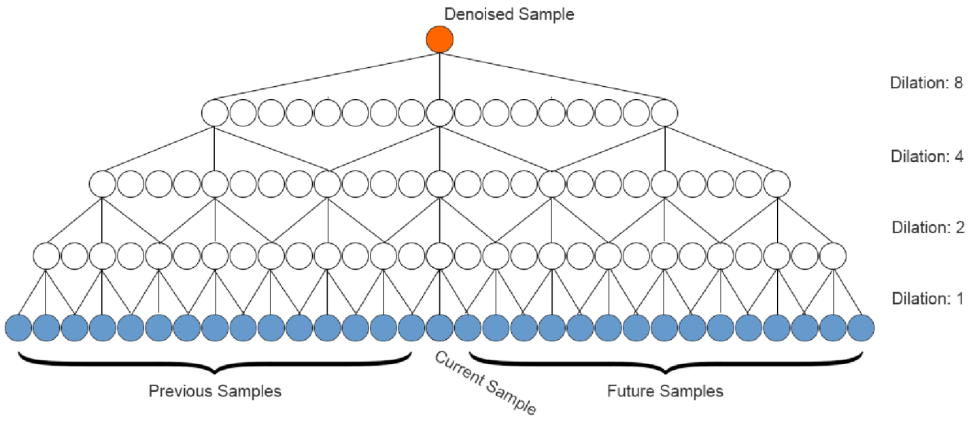


Figure 16: Stacked non-causal, dilated 1d convolution [41]

2.2 Proposed fully convolutional neural network for speech enhancement

2.2.1 Dilated 2-d and 1-d frequency convolution

The input to the CNN consists of a limited number of successive spectrogram frames. However, the frequency dimension is usually several hundreds and requires a larger size of receptive field in order to exploit the contextual information. Thus, as a common practice, it is necessary to increase the size of the filters in the frequency dimension, e.g., Fu *et al.* [8] used a filter with size of 25 in frequency in their implementation.

As discussed before, dilated convolution has been applied in various contexts including imaging segmentation [61] and speech synthesis [51]. In dilated convolution, whenever a filter weight is applied to the input, a fixed number of input values are skipped, which makes the size of receptive field larger than that of the filter. Stacking dilation convolution results in an exponential expansion of the receptive field. In WaveNet, dilation is heavily employed to capture the signal in time domain with massive samples, however, for frequency-domain CNN, the actually input is often a context window of a spectrogram that contains several frames in order to consider the time-domain contextual information, and each frame is a vector that usually consists of a few hundred frequency bins. Since the dimension of frequency is significantly larger than that of time, dilation is only needed at frequency axis to incorporate the frequency information. Hence, we introduced frequency-dilated 2-d convolution, which is shown in Fig. 17

By stacking this dilated convolution with an increasing dilation factor, one could keep the size of filter relatively small, while obtaining a large receptive field in frequency. For example, stacking 7 layers of such a frequency-dilated convolution with a filter size of 3 and dilation of 1, 2, 4, 8, 16, 32, 64 will enlarge the receptive field

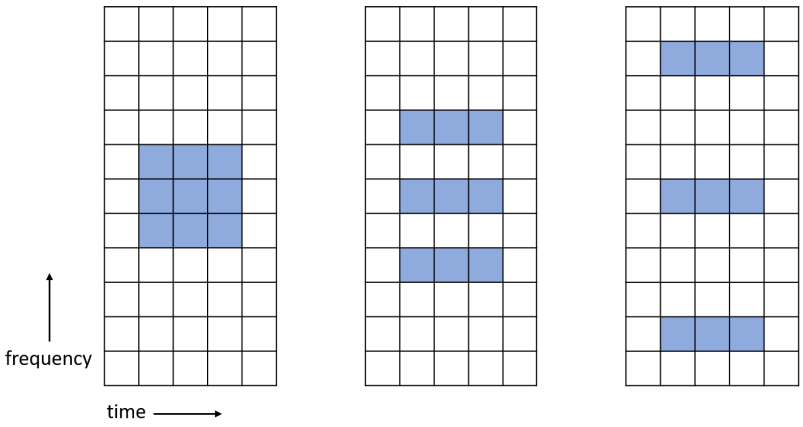


Figure 17: Frequency dilated 2-d convolution

correspondingly, with its size being 3, 7, 15, 31, 63, 127, 255 for each layer. Naturally, using a large filter with dilation of this kind can increase the size of receptive field even more rapidly, for example, when the filter size is increased to 17, stacking 4 layers will eventually produce a receptive field of size 241.

On the other hand, the proposed network also adopts 1-d convolution, which is a special case where the filter is only of 1-dimension. In Fig. 18, the 1-d convolution is applied along frequency axis. It is more efficient than 2-d convolution when the goal is to increase the size of receptive field along frequency axis only. Hence it has been used in some recent works [35, 50]. A special type of 1-d convolution is 1×1 convolution, where the filter size is simply 1×1. Although it does not increase the size of the receptive field, it has been commonly used for channel adjustment, because of its small kernel size which is highly efficient.

2.2.2 Network architecture

Inspired by WaveNet [51], here we propose a convolutional network for complex spectrogram estimation as shown in Fig. 19. It is fully convolutional and consists of a set of 2-d convolutional layers (denoted as Conv2d) and 1-d convolutional layers (Conv1d). The Conv2d layer uses both frequency-dilated 2-d convolution and regular

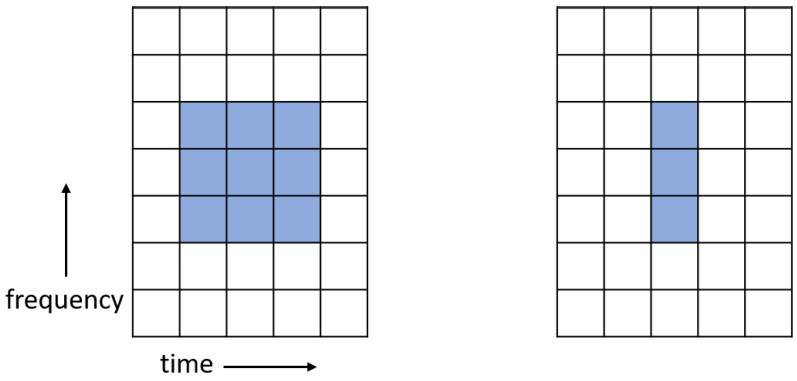


Figure 18: 2d convolution an 1d convolution

1-d convolution, while the Conv1d layer only uses regular 1-d convolution along the frequency axis. It is possible to combine real and imaginary spectrograms as the input [8], yet we find treating them separately may lead to a better performance.

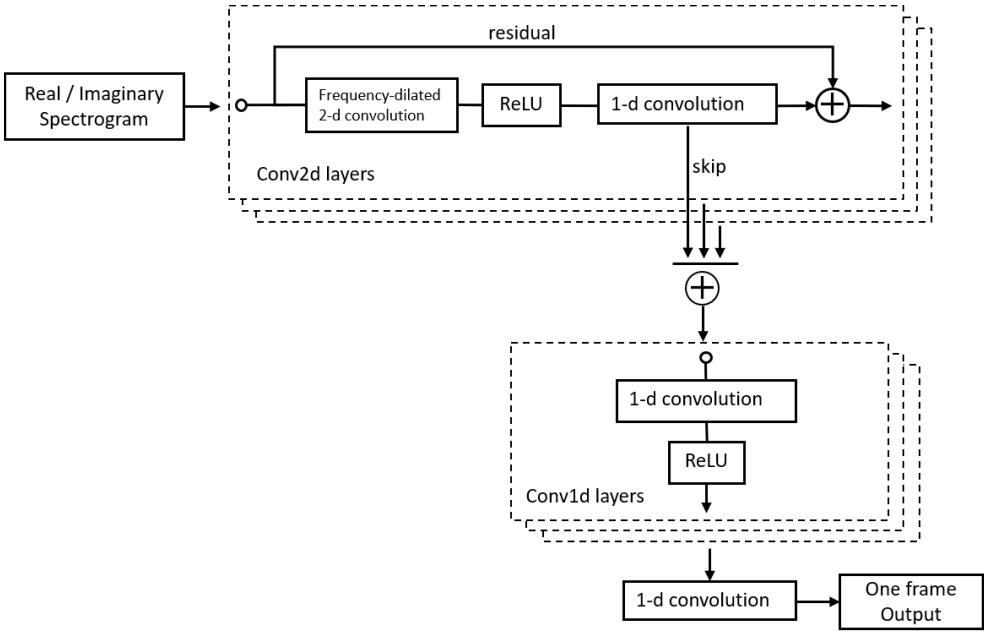


Figure 19: The network structure of the proposed CNN

The input to the CNN consists of 13 frames. Stacking Conv2d layers allows the receptive field of time to increase linearly. When the size of the receptive field along time axis equals or exceeds the number of input frames, the central frame of the output of Conv2d layers will contain the information from all input frames. Hence it

is extracted as the input to Conv1d layers to produce a single-frame output.

Table 3 shows a configuration of the proposed CNN. The Conv2d layers are stacked 6 times with frequency dilation increased by a factor of 2 (i.e., 1, 2, 4, 8, 16, 32), which yields a receptive field of size 253 in frequency and size 13 in time. The Conv1d layers are stacked 2 times, followed by the output layer, which is simply 1-d convolution for different channels (denoted as 1d-real and 1d-imag in the table) to separately produce the real-valued frame and the imaginary-valued frame.

Similar to the WaveNet, in our proposed network, the Conv2d layers also adopt a residual learning and skip-connection structure [15, 51] to ease the training of a deeper network. The residual path provides the next Conv2d layer with lower dimensional data from the previous layer which may be destructively compressed during the convolution process [35], while the skip connection provides the Conv1d layer with the data processed at the current Conv2d layer.

Table 3: The Network Configuration (Config.1). The height of the filter is the size along the frequency axis, and the width is the size along the time axis. The channel refers to the depth, or the number of feature maps of convolution.

Layer name	Filter name	Height	Width	Channel
Conv2d	dilated 2d	5	3	48
	1d-skip	1	1	48
	1d-residual	1	1	48
Conv1d	1d	3	1	96
Output	1d-real	3	1	1
	1d-imag	3	1	1

2.3 Performance evaluation

2.3.1 Performance measure of speech quality

Experiment setup

To evaluate the performance of the proposed model in terms of speech intelligibility, we conducted experiments using TIMIT database [9], in which 780 utterances from both female and male speakers are used for the training and 90 utterances used for testing. Four typical non-stationary noises (babble, street, factory and restaurant) are randomly truncated and used for both training and testing stages. The sampling rate is set to 16 kHz. The SNR levels for training and testing stages are set to -5 dB, 0 dB, 5 dB and 10 dB. We evaluate the proposed CNN towards two aspects: speech quality, and the model complexity.

Comparison with previous models

The proposed CNN is compared with two other complex-spectrogram processing methods: CIRM [58] and RI-CNN [8]. As introduced before, CIRM is a DNN-based method that estimates a complex mask from a set of spectral features. RI-CNN is a CNN model that consists of convolution layers and fully-connected layers, and takes complex spectrogram as input.

For a fair comparison, all networks are trained and tested with the database and the SNR level described above. All models use 500-point DFT with 50% overlap. Apart from the DFT length, both reference methods are implemented with the configuration described in the original papers, which makes the number of parameters for RI-CNN, CIRM and the proposed CNN to be $775K$, $3.87M$ and $243K$, respectively. Aforementioned SSNR and PESQ are used as performance metrics. Table 4 shows the result obtained from each network. Clearly, the proposed CNN outperforms RI-CNN, while achieving a comparable performance to CIRM but with around 16 times fewer

of parameters ($243K$ parameters of proposed CNN compared with $3.87M$ parameters of CIRM).

Table 4: PESQ and SSNR score of different models

metrics	**PESQ**				**SSNR**			
SNR	**-5 dB**	**0 dB**	**5 dB**	**10 dB**	**-5 dB**	**0 dB**	**5 dB**	**10 dB**
unprocessed	1.337	1.678	2.043	2.403	−12.627	−8.901	−4.373	0.274
CIRM	1.824	2.279	2.649	2.956	**−1.137**	**1.332**	3.914	5.959
RI-CNN	1.760	2.115	2.421	2.616	−3.094	−0.584	1.601	3.409
proposed	**1.951**	**2.349**	**2.672**	**2.972**	−1.786	1.255	**4.301**	**7.052**

Benefit to phase processing

To further investigate whether complex spectrogram processing is beneficial to phase estimation, we combine the clean magnitude with either noisy phase or estimated phase from estimated complex spectrogram to synthesize the speech. We have compared the proposed CNN with the two other complex-spectrogram processing methods. The average PESQ scores for both female and male speech are shown in Fig. 20.

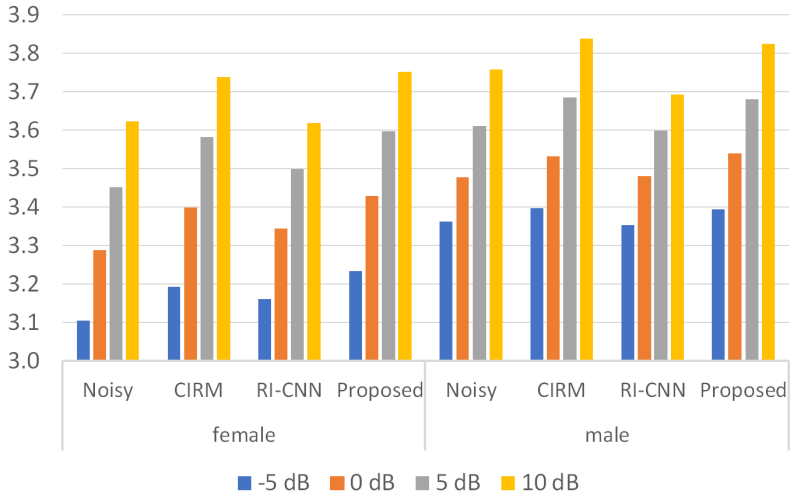


Figure 20: Average PESQ score on female and male speech by replacing phase of noisy speech with that of estimated clean speech.

Again, the proposed method shows a comparable performance with CIRM. RI-CNN is found to be the least effective in phase processing with complex spectrogram estimation. For female speech, a maximal improvement of 0.15 is observed. Yet for male speech, the improvement is less significant, possibly indicating that the perceptual quality of female speech is more prone to phase distortion than that of male, and thus benefited more from phase processing.

It is worth mentioning that, in terms of phase processing, current complex spectrogram estimation algorithms may be less effective than algorithms like [23], which only and directly estimates phase of clean speech and could improve the PESQ score by merely using the estimated phase with the spectral magnitude of noisy speech for reconstructing the estimated clean speech. For complex spectrogram estimation algorithms, however, when using the combination of noisy magnitude and estimated phase from the processed complex spectrogram, the improvement on PESQ is rather limited.

Performance measure with restricted model complexity

Some recent works that utilize CNN for speech processing have considered a situation where the number of parameters is limited [35, 46]. Thus in the third experiment, we have configured the model in a parameter-controlled manner. In addition, the memory footprint of the proposed CNN is also considered. While it is rather implementation dependent, a rough measure could be *(size of spectrogram)*×*(2d convolution channel)*×*(1d convolution channel of Conv2d layers)*×*(size of float)*. Two configurations of the proposed CNN with parameters at the level of $100K$ and $50K$ are tested for the overall denoising performance.

By stacking 6 Conv2d layers and 2 Conv1d layers, the configuration shown in Table 3 has $243K$ parameters, and the memory footprint is around 29 megabytes (MB). Meanwhile, two configurations shown in Table 5 and Table 6 keep the number

Table 5: A network configuration where the number of parameters is $97K$ (Config.2)

Layer name	Filter name	Height	Width	Channel
Conv2d	dilated 2d	5	3	32
	1d-skip	1	1	24
	1d-residual	1	1	24
Conv1d	1d	5	1	64
Output	1d-real	17	1	1
	1d-imag	17	1	1

Table 6: A network configuration where the number of parameters is $50K$ (Config.3)

Layer name	Filter name	Height	Width	Channel
Conv2d	dilated 2d	5	3	32
	1d-skip	1	1	16
	1d-residual	1	1	16
Conv1d	1d	1	1	48
Output	1d-real	17	1	1
	1d-imag	17	1	1

of Conv2d and Conv1d layers unchanged, but use fewer filter channels to reduce the number of parameters and the memory footprint at the same time. The config.2 shown in Table 5 has $97K$ parameters, and the memory footprint is 10 MB. The config.3 in Table 6 further reduces the parameters and the memory footprint to $50K$ and 6 MB, respectively. Figure 21 illustrates the overall performance for all three configurations. While the model with $97K$ parameters still produces a good overall result, the one with $50K$ suffers more loss on PESQ score. Generally speaking, config.2 seems to reach a good balance between the denosing performance and memory efficiency.

2.3.2 Performance measure of keyword spotting

An application of speech enhancement is to improve the performance of speech recognition system under noisy conditions. In this case the speech enhancement method

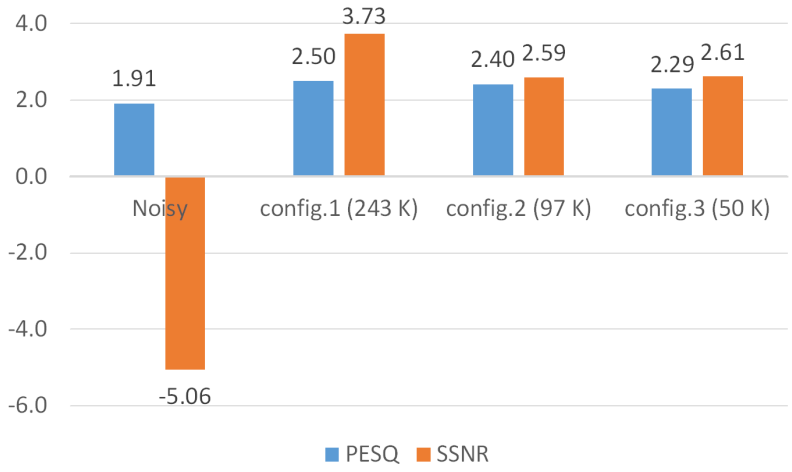


Figure 21: Performance comparison with different model configurations

could be considered as a pre-processor. It has been reported that the traditional objective of speech enhancement may not be consistent with the objective of speech recognition, i.e., a better speech quality doesn't necessarily result in a better speech recognition performance. Empirically, SSNR is preferred over PESQ if no prior knowledge of the speech recognition algorithm is obtained.

To evaluate whether the proposed CNN is applicable for SE for speech recognition, we use a keyword spotting engine that detects keyword *Alexa*. The engine is assumed to be a black-box, i.e., no prior knowledge is given except that it will output a stream of boolean decision where *true* indicates a keyword has been detected and *false* otherwise. We use two metrics for the evaluation, namely false-reject rate (FRR) and false-alarm rate (FAR). FRR refers to the number of keywords being ignored by the engine given a stream of keyword and FAR refers to the number of occurrences that the engine detects a keyword when the input is some other sound. In addition, we train the same model on various datasets and demonstrate that the choice of training dataset can be crucial as stated in Chapter 1.

Experiment setup

To measure FRR, we use a speech dataset that contains recordings with keyword *Alexa*. Fig. 22 shows the recording environment and the speaker layout for the keyword speech dataset. Recording is done in an approximately 4.8m×5.8m room with carpeted floor, plaster walls, and drop tile ceiling. Mouth speakers refers to the speaker devices for playing speech audio and are placed at 2m and 4m radii and 60, 90, and 120 degrees relative to device under test (DUT). Noise speakers refers to the speaker devices for playing the noise audio and are placed at 2.5m and 45, 135 degrees relative to DUT. For one test recording, two noise speakers are playing the noise audio simultaneously while only one month speaker is playing the speech audio at one distance and angle. The whole test dataset covers all six mouth speakers at different distance and angles.

Since both speech and noise are captured by the microphone at the same time, it is rather difficult to calculate the SNR level accurately. Hence, the SNR level is roughly divided into 5 levels: -6 dB, -3 dB, 0 dB, 3 dB and 6 dB according to the noise gain relative to 56dBA reference level. For each SNR level, the test recording contains 4.6 h of audio covering all mouth speaker positions shown in the figure.

For FAR evaluation, in addition to the keyword recording dataset, we use a BBC recording dataset which contains 24 h of BBC recording without the appearance of keyword Alexa.

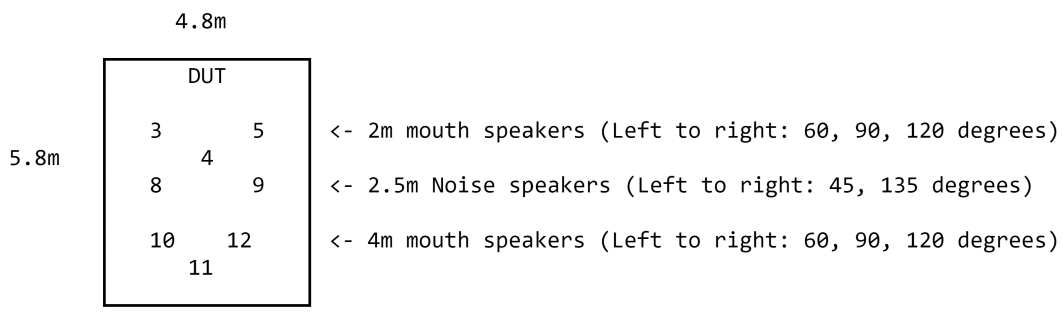


Figure 22: Speaker layout of the KWS recording dataset

For comparison, the DNN-based CIRM method and proposed model are trained on TIMIT dataset and evaluated for their performance on both FAR and FRR. In addition, in order to investigate the effect of training speech dataset on the KWS performance, the proposed model is also trained on various speech datasets. We use 4 speech datasets: a keyword set that is obtained from the keyword dataset for testing, IEEE dataset that contains 720 utterances from one male speaker, TIMIT dataset contains 6300 utterances from 630 speakers and TED-LIUM (version 2) dataset that contains more than $1k$ TED talks from more than 600 speakers.

Experimental results

Table 7, 8 and 9 show the results on FAR and FRR for DNN-based CIRM and the proposed method. It is apparent that the proposed model is preferred over CIRM in terms of FRR. Regarding the FAR, the proposed method produces a certain number of FAs on BBC recording dataset while the CIRM yields no FA at all. On the other hand, both method produce no FA on the keyword recording dataset.

Table 7: Number of misses per 60 keywords of the proposed model and CIRM

	-6 dB	**-3 dB**	**0 dB**	**3 dB**	**6 dB**
unprocessed	56.2	46.8	33.3	21.3	10.5
CIRM	46.4	37.3	30.1	20.6	14.5
Proposed	**42.3**	**29.8**	**17.4**	**9.3**	**4.8**

Table 10 and 11 display the FRR and FAR of the proposed model trained on different speech datasets, respectively. Obviously, the training dataset plays an important role in the final KWS performance. Interestingly, the model trained on TED-LIUM dataset is preferred over the model trained on the keyword set, which is directly a subset of test data. Table 12 shows the FAR result on 24h-BBC recording dataset. The model trained on IEEE dataset yields a plenty of false alarms while the model

Table 8: Number of false alarms per hour BBC recording of the proposed model and CIRM

	FAR
unprocessed	0
CIRM	0
Proposed	0.208

Table 9: Number of false alarms per hour keyword recording of the proposed model and CIRM

	-6 dB	**-3 dB**	**0 dB**	**3 dB**	**6 dB**
unprocessed	0	0	0	0	0
CIRM	0	0	0	0	0
Proposed	0	0	0	0	0

trained on the keyword set and TED-LIUM produce a fairly small number of false alarms. The model trained on TIMIT dataset, on the other hand, produces no false alarm at all.

Overall, the training dataset has a considerable impact on the SE performance for KWS task. TIMIT dataset and TED-LIUM may be solid options for training the model if the objective is to improve KWS performance.

Table 10: Number of misses per 60 keywords of proposed model with different training speech datasets

	duration	**-6 dB**	**-3 dB**	**0 dB**	**3 dB**	**6 dB**
unprocessed	-	56.2	46.8	33.3	21.3	10.5
keyword set	5 m	**38.7**	**27.3**	17.7	12.1	9.1
IEEE	32 m	41.8	31.2	22.5	14.4	8.4
TIMIT	5 h	42.3	29.8	17.4	9.3	4.8
TED-LIUM	207 h	40.8	27.5	**15.2**	**7.2**	**2.8**

Table 11: Number of false alarms per hour keyword recording of proposed model

	duration	-6 dB	-3 dB	0 dB	3 dB	6 dB
unprocessed	-	0	0	0	0	0
keyword set	5 m	0.435	0	0	0	0
IEEE	32 m	10.435	3.696	0.217	0.435	0
TIMIT	5 h	0	0	0	0	0
TED-LIUM	207 h	0.652	0.217	0	0	0

Table 12: Number of false alarms per hour BBC recording of proposed model

	duration	Totoal FA
unprocessed	-	0
keyword set	5 m	0.208
IEEE	32 m	1.5
TIMIT	5 h	0.208
TED-LIUM	207 h	0.417

2.4 Summary

In this chapter, a fully convolutional network for complex spectrogram estimation in single-channel speech enhancement is proposed. First, some previous works on neural networks based speech enhancement, including fully-connected DNN based and CNN based methods, are briefly introduced. Useful CNN structures such as convolutional encoder-decoder and WaveNet are also introduced. The proposed network is then described in detail. The network adopts skip-connect and residual learning, and features frequency-dilated convolution for spectrogram processing, which yields an exponential growth of the receptive field in frequency domain. The proposed network structure can lead to an efficient implementation that requires fewer parameters as compared with conventional fully-connected DNN and CNN while still producing a comparable performance.

The experimental results show that the proposed method performs better than the existing methods in terms of speech quality with a much smaller model size, and is suitable for speech enhancement for keyword spotting. It is also observed that the choice of training dataset has a large impact on the performance of model when the objective is SE for keyword spotting task.

Chapter 3

Signal-Channel Speech Enhancement with Generative Adversarial Network

In this chapter, we develop a method for single-channel speech enhancement using generative adversarial network (GAN). This chapter is organized as follows. In Section 3.1, background and recent works in speech enhancement using GAN are introduced. Section 3.2 introduces the proposed model for single-channel speech enhancement.The performance of the proposed method is evaluated with comparison to some existing works in Section 3.3.

3.1 Previous work

3.1.1 Generative adversarial network

Introduced by Goodfellow et al., generative adversarial networks (GANs) have gained major interest for their capability of modeling the underlying data distribution through adversarial training. GAN refers to a generative model that consists of a generator

(G) and a discriminator (D) and utilizes unsupervised adversarial training with the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{7}$$

where x represents the data samples from a given training dataset, z represents some latent noise vector sampled from some known prior probability distribution. As illustrated in Eq. 7 and Fig. 23, the training process is a min-max game between G and D. Both G and D are trained simultaneously. The G maps the latent vector z to the data sample x, while the D determines whether the input to D is the data sample x from the training dataset (often denoted as *real*) or generated by G (denoted as *fake*).

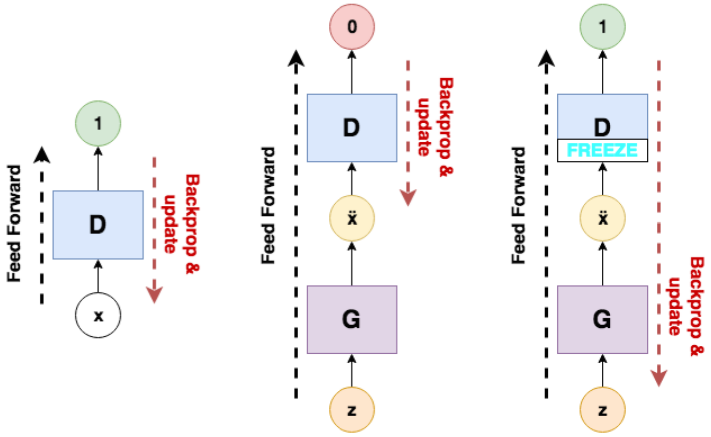


Figure 23: The training process of GAN. The gradient on training loss on real data samples that come from the training dataset and fake samples that generated by G are back-propagated on D. The parameters of D are then frozen to compute the training loss on G and its gradient are back-propagated to adjust G's parameters. [36]

The adversarial training of GAN is known as notoriously difficult. Least squares GAN (LSGAN) is proposed to stabilize the training process by replacing the sigmoid cross-entropy loss with the least squared loss in order to solve the vanishing gradient problem, i.e.,

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[(D(\mathbf{x})-1)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[D(G(\mathbf{z}))^2] \tag{8}$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[(D(G(\mathbf{z}))-1)^2] \tag{9}$$

Regarding its application, GAN has been heavily investigated in the filed of computer vision (CV) for generative tasks such as image synthesis. Radford et al. demonstrated that a deep convolutional GAN (DCGAN), which features a fully-convolutional structure with a stack of 2d-transposed convolution (sometimes called fractionally-strided convolution) with decreasing depth as shown in Fig. 24, can generate realistic images such as bedroom photos when trained on the corresponding dataset. Donahue et al. further shows the applicability of DCGAN for general audio synthesis task such as generating ambient sound and speech as shown in Fig. 25, where the DCGAN is employed to generate the log-magnitude spectrogram of an audio and then the iterative Griffin-Lim algorithm [14] is applied to estimate the corresponding spectral phase and to produce audio samples with a duration of 1-second. The limitation of audio synthesis DCGAN is rather obvious: since the input is a latent noise vector, the generated samples tend to be very random and consist of low-quality samples which are often unrecognizable to human. Meanwhile, as the model does not include any conditions, the output of the network could be in any genre if the training dataset consists of audio samples with different categories of sound.

3.1.2 The application of GAN in speech enhancement

Recently, researchers have introduced GAN to the field of SE. Specifically, the conditional GAN (CGAN) [31] has been employed for SE task as a general framework in many works [6, 27, 30, 36, 56] due to its ability of generating desired output conditioned on the additional information fed into the network. The objective function of

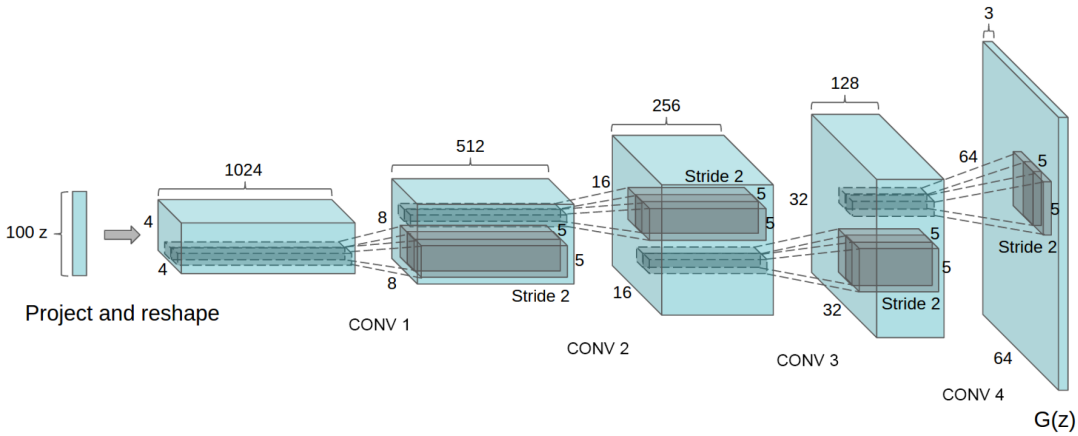


Figure 24: The generator of DCGAN uses transposed 2d convolutions. The input z is a latent vector sampled from a uniform distribution. The output is a RGB image (3 channels) with a resolution of 64×64[40]

CGAN is defined as:

$$\min_G \max_D V_{CGAN}(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z}|\mathbf{y})))] \quad (10)$$

where $\mathbf{y}$ is the extra information of any modalities, which is fed into both D and G as additional input, and $\mathbf{x}$ and $\mathbf{z}$ represent the sample from the training dataset and the latent noise input, respectively. The structure of CGAN is shown in Fig. 26. Straightforwardly, the generator of CGAN is trained to generate samples based on the given information $\mathbf{y}$ and the discriminator is trained to distinguish if the input is real or fake based on the extra information $\mathbf{y}$. The modality of the additional information, on the other hand, has to be determined when designing the network. In the context of image synthesis, the extra information can be the label of output image. For example, in [31], the author demonstrated CGAN can be trained to generate images of hand-written digits from MNIST dataset [25], where the extra information is simply one-hot encoded label of the digits, and the results are shown in Fig. 27.

In the context of SE, the additional information is simply noisy speech in either time-domain [36] or frequency-domain [6, 27, 30, 56]. Fig.28 shows the structure of

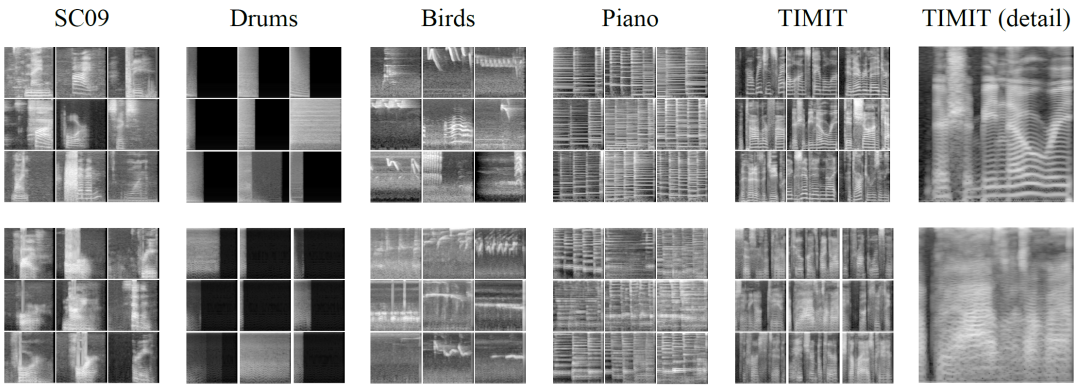


Figure 25: The magnitude spectrogram of audio samples from five dataset. **First row:** audio samples from training dataset. **Second row:** audio samples generated by GAN [7]

SEGAN, a CGAN-based SE method that directly processes speech in time domain. Fig. 29 reveals the design of a variant of SEGAN which is called FSEGAN and operates in frequency domain through magnitude spectrogram estimation. In [30] a CGAN framework called Pix2Pix [19] is employed to enhance speech in the frequency-domain. Intuitively, G estimates the magnitude spectrogram of clean speech while D tries to identify whether the input spectrogram is enhanced or clean, both conditioned on the noisy spectrogram of speech. While CGAN can be directly adopted for SE without major modifications [30, 36], some researchers use a variant of CGAN for SE in which the random noise input is removed to obtain a deterministic model [6, 27, 56].

Regarding the specific network implementation, a similar design as illustrated in Fig. 30 has been shared in multiple works [6, 27, 30]. While the specific implementation of G, such as the size of filters, the number of feature maps, etc., may vary, its architecture follows an encoder-decoder design featuring a symmetric convolutional structure, where the encoder outputs some compressed, high-dimensional representation of the input data and the decoder reconstructs the data based on the representation. On the other hand, the D could be considered as a combination of an encoder and a classifier. The encoder outputs compressed features, and the classifier

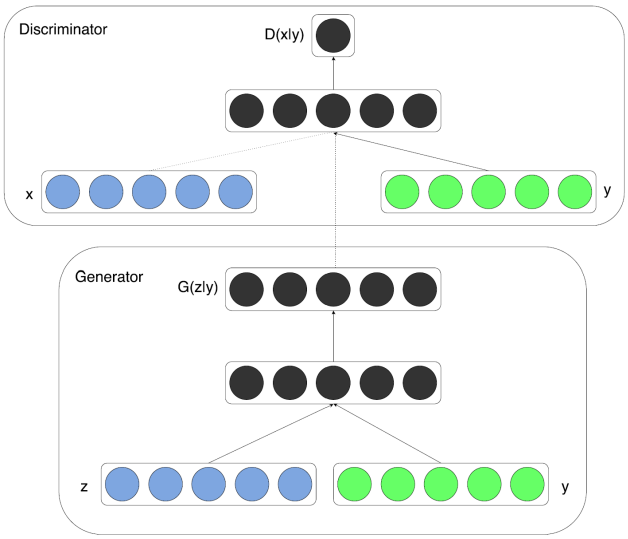


Figure 26: The structure of a simple CGAN [31]

decides whether the input is directly from the training dataset or generated by G based on those features.

3.2 Proposed InfoGAN-based method for speech enhancement

3.2.1 Introduction to InfoGAN

Originally, the InfoGAN is designed for unsupervised representation learning for generative tasks. It is similar to CGAN in the sense that additional information is fed to G as constraints to produce the desired output.

However, instead of specifying the type of additional information (e.g., label of the desired output) when designing the network, InfoGAN can be trained to learn this high-level information of the data by adding a regularization term to GAN's objective function:

$$\min_{G} \max_{D} V_I = V(D, G) - \lambda_I I(c; G(z, c)) \tag{11}$$

where $V(D, G)$ represents the original objective function of GAN, c denotes the latent

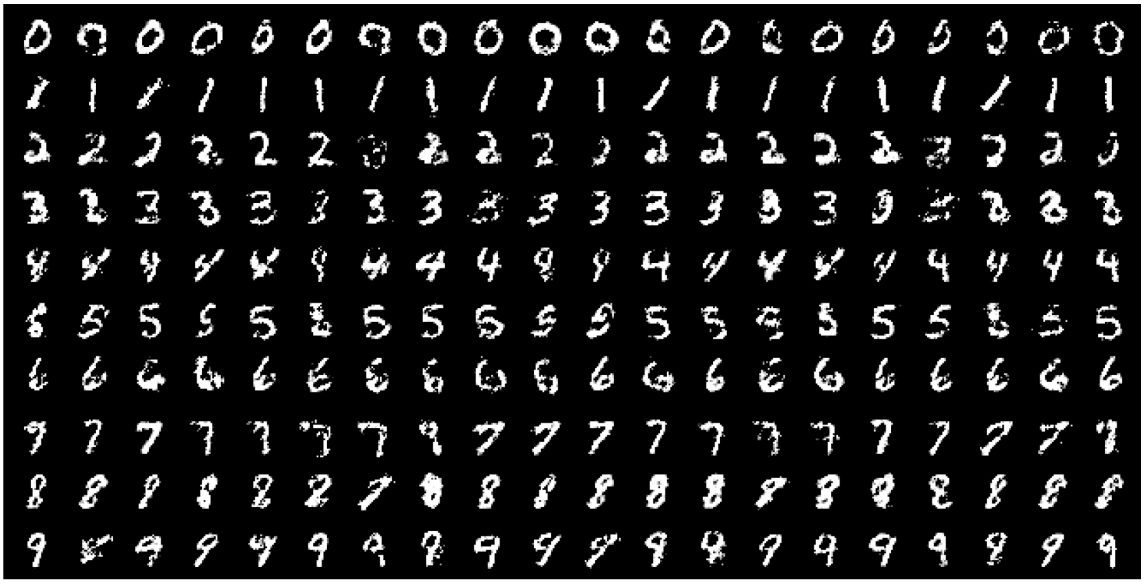


Figure 27: Samples of MNIST digits generated by CGAN. Each row is conditioned on one label [31]

code and z denotes the input noise. The latent code is some high-dimensional data that controls the output of G. Hence, by manipulating the code, G could generate output as desired by users. Controlled by a hyper-parameter λ_I, the term $I(c; G(z, c))$ represents the mutual information between the latent data and the output of G.

Fig. 31 shows the framework of InfoGAN, where the input to G consists of random noise sampled from some prior distribution and a latent code which is sampled from a distribution whose parameter is estimated by an auxiliary network Q appended to D. In practice, the information regularization term is replaced by its lower bound $L_1(G, Q)$ since the posterior $P(c|x)$ (x represents the real data from training dataset) is required to calculate the term yet hard to obtain:

$$\begin{aligned} L_1(G, Q) &= \mathbb{E}_{\mathbf{c}\sim P(\mathbf{c}),\mathbf{x}\sim G(\mathbf{z},\mathbf{c})}[\log Q(\mathbf{c}|\mathbf{x})] + H(\mathbf{c}) \\ &= \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{c})}[\mathbf{c'} \sim \mathbb{P}(\mathbf{c}|\mathbf{x})[\log Q(\mathbf{c'}|\mathbf{x})]] + H(\mathbf{c}) \\ &\leq I(\mathbf{c}; G(\mathbf{z}, \mathbf{c})) \end{aligned} \tag{12}$$

Eventually, with the lower bound approximation, the objective function is defined

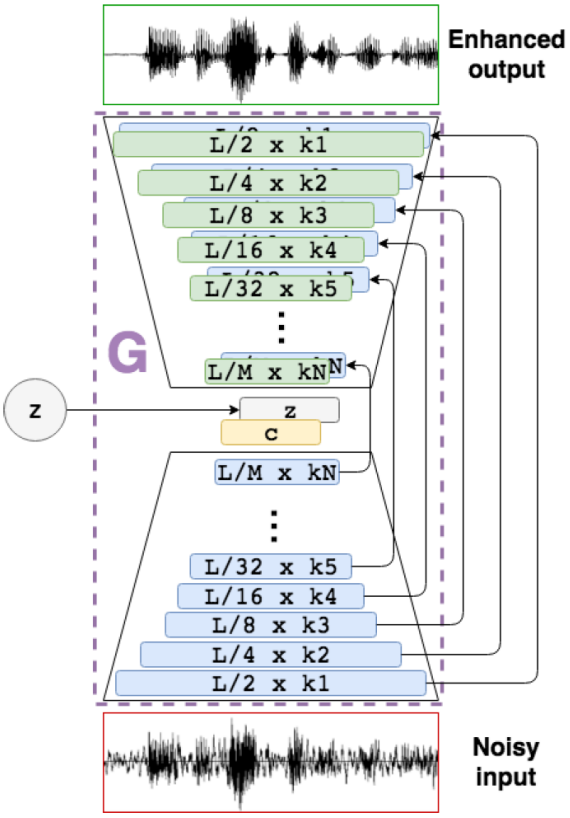


Figure 28: The network structure of SEGAN [36]

as:

$$\min_{G,Q} \max_{D} V_I = V(D,G) - \lambda_I L_I(G,Q) \tag{13}$$

It is trivial that, the mutual information term $I(c; G(z,c))$ approaches 0 when c and $G(z,c)$ are independent, and the term reaches its maximum when c and $G(z,c)$ are related in a deterministic, invertible way. The interpretability of the latent code, on the other hand, largely depends on the form of distribution that the latent data is modeled with, as well as the representation that the latent code tries to disentangle. While the mutual information term sometimes does make the latent code lean towards a high-level representation that human can interpret and manipulate, there is no guarantee that it can always be understood by human, because the semantic modality of the data is often the straightforward for human but is not strictly related to the

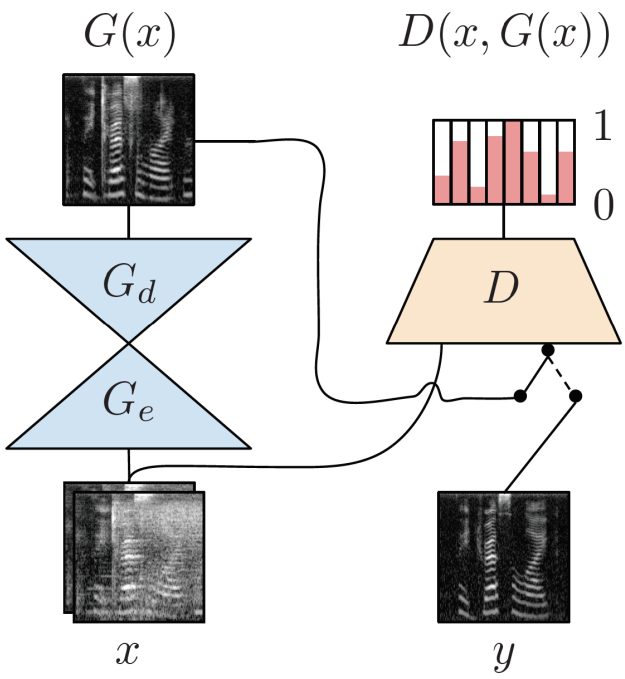


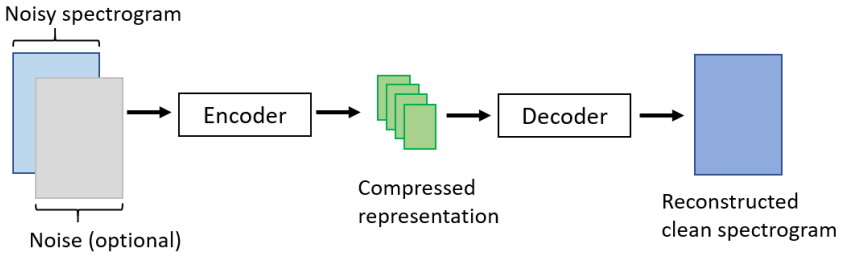
Figure 29: The network structure of FSEGAN [6]

mutual information term.

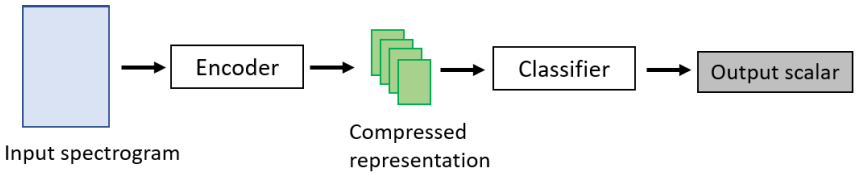
While InfoGAN is shown to work fairly well on certain datasets that consists of clean images without many varieties on the characteristics, such as hand-writeen digit dataset MNIST and 3D chair image dataset [3], its capability of disentangling the features may be rather limited on more complex datasets such as car dataset [22], where characteristics of the photos of cars form a large collection which is overwhelmingly difficult to model with any probability distribution. Moreover, as the framework of InfoGAN does not accept any predefined characteristics, it is unsure which feature or characteristic that the InfoGAN will be trained to disentangle.

3.2.2 Adopting InfoGAN for speech enhancement

Several recent works [7, 27, 30, 36, 56] have developed GAN-based SE methods, which can produce promising results. These methods are also known to have a very high



(a) Structure of G



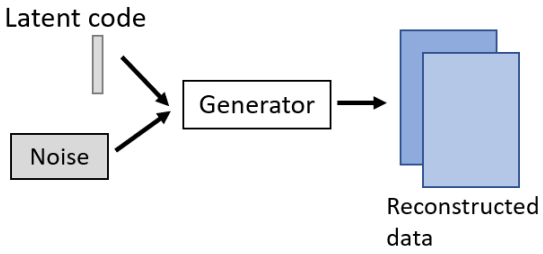
(b) Structure of D

Figure 30: A common GAN architecture for SE task

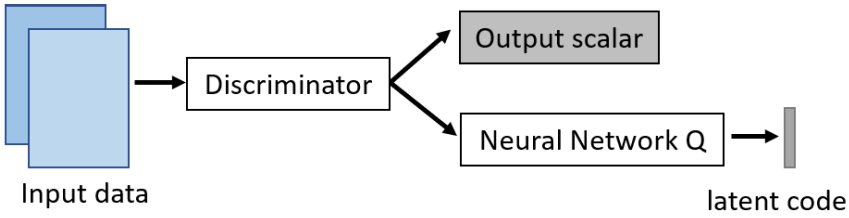
complexity. For example, the CNN-based GANs for SE are usually configured with millions of parameters leading to an even larger computational cost.

To address this problem, we investigate information maximizing generative adversarial network (InfoGAN) [5] for SE task, which is capable of producing a high-level representation of data. It is known that D is usually employed only in the adversarial training stage and is not used in the speech-enhancing stage while G serves as a direct mapping function to enhance speech by adopting an encoder-decoder design. In the proposed method, however, both D and G are involved in the speech-enhancing stage without requiring additional computations. We design D to be an encoder-like network and G to be a decoder-like network, which consequently results in a more compact system. To further improve the efficiency, we extend D's functionality as a SNR indicator without additional cost, and show that the proposed method could save the system from redundant computations by detecting the clean or near-clean speech which does not require further processing.

Speech enhancement is a regression task by its nature [39]. As the relation between noisy and clean speech is often deterministic, it is straightforward to employ a neural



(a) The generator of InfoGAN



(b) The discriminator of InfoGAN

Figure 31: The InfoGAN framework

network as a explicit mapping function working in a regression approach as adopted in most works, which consequently, requires supervised training. In fact, most deep-learning-based methods are designed in this approach.

Since GAN is essentially a generative model that features unsupervised training, some common practice has been employed in many SE works to obtain a deterministic model with supervised training. Conventionally, the random noise is often considered as latent information that enforces the generator to capture the true distribution of the training data by mapping the latent information to some high-dimensional space. However, for SE task, the random noise is often removed from generator's input, given that the mapping function which G serves as only relates the noisy speech and the clean speech, and the input speech to the system is already noisy which requires no hyper-parameters to play a part in the adversarial training.

On the other hand, the $\ell 1$ norm or the mean-square-error (MSE) of the difference the generator's output and the clean speech is often added as an extra loss-component to speed up the training. It has been shown that, for specific architectures of CGAN,

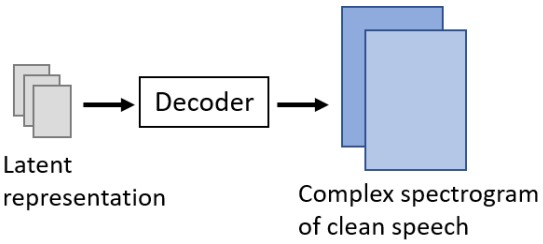
such as Pix2Pix [19], $\ell 1$ distance may be preferred over MSE because it leads to less blurring among the output of G and tends to generalize better.

While it is possible to adopt InfoGAN with unsupervised training or semi-supervised training, where a generative model can be trained only on clean speech and then be employed to generate the clean speech that best matches the context of noisy speech (as described in Chapter 1), we attempt to adopt the InfoGAN framework in a deterministic manner with supervised training to take advantage of the more sophisticated methodology that has been explored and discussed in numerous works.

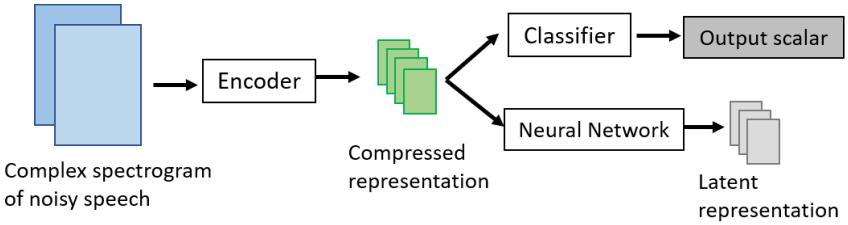
SE does not require the model to produce a disentangled, meaningful representation of the speech itself (which is more related to speech-text translation), as a result, the InfoGAN does not require unsupervised training to interpret the context of speech, which largely simplify the task and consequently the training process. With supervised training, the InfoGAN creates an explicit mapping between the noisy speech and its latent representation, and between the latent representation and the estimate of clean speech. Since the mapping function is deterministic but not necessarily invertible, the mutual information between the latent representation and the output of generator is expected to be high.

Although InfoGAN's capability of generating interpretable representations is not very useful for SE task, we make use of the design of its system framework and architecture in the context of SE. Fig. 32 shows our proposed InfoGAN-based SE method. As opposed to the conventional GAN design where G is in the form of an encoder-decoder, the G here simply works as a decoder that reconstructs the clean speech using a latent representation of noisy speech at the output of D.

As suggested by [6], the input noise to the generator may be redundant given the presence of noise in the input spectrogram, and thus it is removed in our work. The objective function of the proposed SE InfoGAN is defined in a least-squared [29] form:



(a) The generator of InfoGAN for SE



(b) The discriminator of InfoGAN for SE

Figure 32: A InfoGAN framework for SE

$$\min_D V(D) = \frac{1}{3}\mathbb{E}_{\mathbf{x}\sim noisy}[(D(\mathbf{x}))^2] + \frac{1}{3}\mathbb{E}_{\mathbf{y}\sim clean}[(D(\mathbf{y})-1)^2] + \frac{1}{3}\mathbb{E}_{\mathbf{c}\sim latent}[(D(G(\mathbf{c})))^2] \tag{14}$$

$$\min_G V(G) = \mathbb{E}_{\mathbf{c}\sim latent}[(D(G(\mathbf{c}))-1)^2] + \lambda\mathcal{L}_{\text{MSE}}(G(\mathbf{c}),\mathbf{y}) \tag{15}$$

where x, y and c denote the noisy spectrogram, clean spectrogram and latent representation of the input speech, respectively. Since the latent data is estimated by D deterministically, the objective function does not include the term that maximizes the mutual information. An additional term is added to minimize the mean square distance between the clean spectrogram and its estimate, and it is controlled by a hyper-parameter λ, which is set to 1 in our experiment.

In contrast to the conventional GAN-based SE methods, in our proposed SE InfoGAN model, D is involved not only in training but also in speech-enhancing stage to produce c, the latent representation, as the input of G. To take advantage of this procedure, we add a term $\mathbb{E}_{\mathbf{x}\sim noisy}[(D(\mathbf{x}))^2]$ to the objective function. This enables D to work like an SNR indicator. Ideally, the value of the output scalar of D

is near 0 when the speech is very noisy, and gets closer to 1 as the speech tends to be more clean.

The system diagram of the proposed SE InfoGAN is given in Fig. 33. The input noisy spectrogram is first passed to D to simultaneously estimate the latent representation and the scalar that reflects the SNR of the signal. If the output scalar is larger than a pre-set threshold D_{thre}, which indicates that the signal is rather clean and requires no processing, the signal is left unprocessed as the output of the system. Otherwise the latent representation will be fed to G to estimate the spectrogram of clean speech. In the following section we will demonstrate that this process could save the system from redundant computations.

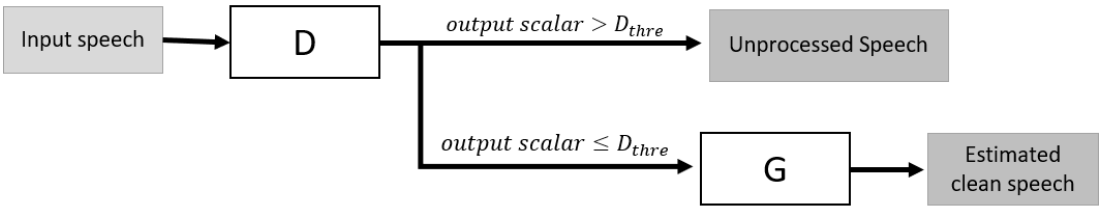


Figure 33: The SE InfoGAN system diagram

The specific network design of both G and D, on the other hand, is very flexible. Recent progress in CNN features residual learning and skip connection [16, 51] which creates shortcut for gradient to flow back to previous layers during back propagation to alleviate gradient vanishing.

To incorporate this structure into the network design, we adopt the redundant convolutional encoder-decoder (RCED) structure [35], where the spatial size of feature maps keeps unchanged but their numbers increase within the encoder and decrease within the decoder. The encoder (D) and decoder (G) could be considered as two separate CNNs that use internal shortcut across layers within each network. The structure of encoder (D) is shown in Fig. 34 (a). The G uses the same structure as D, but the number of feature maps keeps decreasing throughout the network.

Note that it is also possible for the encoder (D) and decoder (G) to adopt 2d convolution (down-sampling) and transposed 2d convolution (up-sampling) respectively [6, 27, 43]. With this design, the shortcut connection is setup across G and D, as illustrated in Fig. 34 (b).

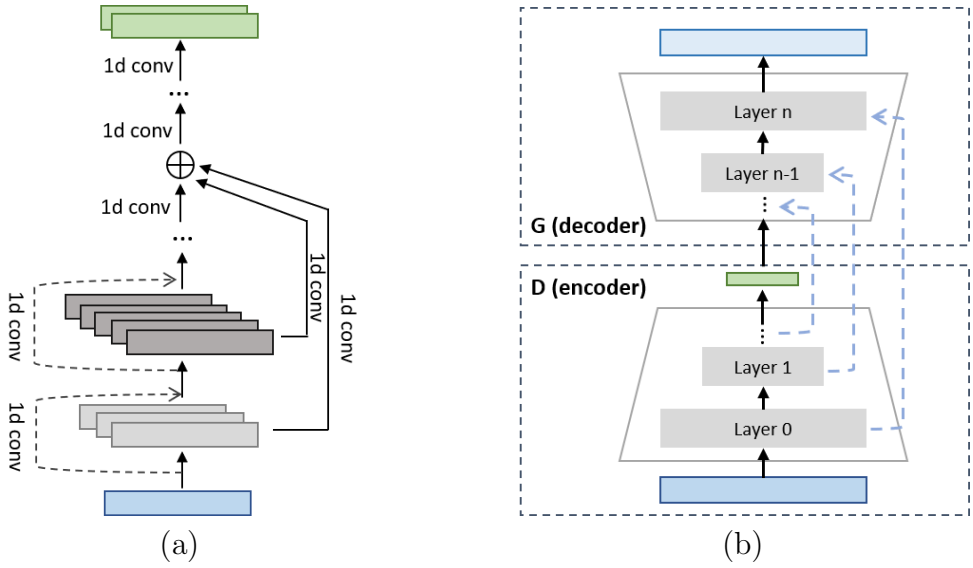


Figure 34: Architectures that incorporate residual learning and skip connection.

3.3 Performance evaluation

3.3.1 Experimental setup

For training, we use TIMIT dataset [9] that consists of $6.2k$ utterances. The noise dataset consists of 50 types of sound from ESC-50 [38] and 6 types of noise from Noisex-92 [53]. The SNR level ranges from -5 dB to 10 dB with 1 dB step increase. The sampling rate is $16k$ Hz. To create a mismatch between training and evaluation, we exclude 100 utterances and several types of unstationary noises (babble, factory, street) from the training data. Both input and output of the proposed system contain 4 frames of complex spectrogram of speech. Short-time Fourier transform (STFT) is applied to obtain the complex spectrogram with a window size of 32 ms and a

hop size of 16 ms. Both real and imaginary spectrograms contain 257 frequency bins. While a 50% overlap is used between frames, there is no overlap between the past and current input data, and equivalently, the corresponding output data, i.e., both input and output consist of 64 ms of non-overlapped speech. Not using overlap between input units increases the system latency and decreases the computational cost.

It is noteworthy that, the system latency, which refers to the minimal delay between input and output that the system could possibly achieve, is often determined by the framing operation and how the frames are gathered as input. Most deep-learning-based SE methods that operate in frequency domain often gather a number of frames and then feed into the network. In this case, the latency depends on the window size, the hop size and whether there is overlap between different input unit (e.g., the past and current input data, not the overlap between frames). For example, for the proposed method the latency is simply 64 ms as no overlap is used between different input units.

On the other hand, while the choice of the number of frames of input and output does have an impact on the value of parameters (specifically, the weights and biases of the filters), it does not relate to the computational complexity given that there is no overlap between the sequential input units. In fact, the computational cost is roughly proportional to the number of parameters and the number of frequency bins per frame. If measured with the number of multiplies for obtaining a fixed duration of speech, the computational cost can be approximately calculated by:

$$\text{number of multiplies} = \frac{\text{number of parameters} \times \text{number of frequeny bins per frame}}{\text{time duration per frame}} \tag{16}$$

3.3.2 Performance measure of speech quality

For the proposed SE InfoGAN, the filter size is set to 5×3 for both G an D (5 on frequency axis and 3 on time axis). The G contains 3 $2d$-convolutional layers with

a decreasing channel number: 64, 32, 16, followed by 3 $1d$-convolutional layers with filter size of 7×1 and the channel numbers are 16, 8, 2. The number of channels for skip connection is set to 32. The D contains 4 $2d$-convolutional layers with increasing channel numbers: $8, 16, 32, 64$. Dilation is used within D on frequency axis with a rate increased by 2's power. The classifier network of D contains a 1d-convolutional layer, followed by a fully-connected layer to produce the single output value. The network for estimating the latent representation is simply an 1d-convolutional layer appended to the encoder. Under this setup (Config. 1), the number of parameters of G and D are $157k$ and $131k$, respectively.

It is noteworthy that the number of parameters not only reflects the space complexity, but also directly relates to the computational complexity. Hence in our experiment we also investigate the performance of the method with limited model size. A straightforward way of tuning model size is to reduce the number of channels across the convolutional layers. In the second model setup, for G network, the channel numbers of convolutional layers are set to 48, 24, 12 for $2d$ convolution and 12, 6, 2 for $1d$ convolution. For D network, the channel numbers are set to 4, 16, 32, 36 for 2d-convolutional layers. This configuration (Config. 2) yields $99.1k$ parameters for G and $55.6k$ parameters for D.

The performance is evaluated in terms of perceptual evaluation of speech quality (PESQ), signal to distortion ratio (SDR) and segmental signal-to-noise ratio (SSNR). For evaluation purpose, we also implement DNN-based CIRM [58], and a common GAN-based SE model employed in [6, 27] which we simply denote as FSEGAN. The DNN-based CIRM uses the same network configuration as in the original paper and contains $3.88m$ parameters. To allow a fair comparison with our work, the FSEGAN has been tested many times to obtain a network configuration that achieves a balance between model complexity and SE performance. The input and output of FSEGAN are complex spectrogram containing 16 frames. The filter size is also 5×3. The G

adopts a symmetric design and contains 4 encoder layers and 4 decoder layers. The encoder layer uses $2d$ convolution with a stride factor of 2, and the channel numbers are 16, 32, 64, 128. The decoder layers use transposed $2d$ convolution with a stride factor of 2, and the number of channels is 128, 64, 32, 2. Skip-connection is employed across corresponding layers. The D contains 4 convolution layers with channel number 32, 64, 128, 256 and a fully-connected layer. The number of parameters of G is $807k$, and that of D is $651k$.

Table 13 shows the results of the models. The proposed InfoGAN-based SE method could produce a comparable result while the model size ($288k$ of Config.1, $155k$ of Config.2) being significantly smaller than the DNN-based CIRM ($3.88m$) and GAN-based SE model ($1.46m$). Similar to common CNN-based SE methods, the proposed model could be configured with limited parameters while avoiding major penalties on the SE performance.

Table 13: PESQ, SDR and SSNR score on different models

	unprocessed	**CIRM**	**FSEGAN**	**Proposed**	
				config. 1	**config. 2**
PESQ	1.87	2.12	2.16	**2.19**	2.14
SDR	2.60	6.10	8.16	**8.18**	6.87
SSNR	−6.25	−0.73	**0.22**	−0.02	−1.22

Note that, while results may show that the proposed method is preferred over CIRM and FSEGAN in terms of speech intelligibility, the difference between the performances of all three method is not significant, which further implies the SE performance of three methods may be at the same level taking into account the bias-variance tradeoff that is discussed in Chapter 1.

Analysis on discriminator

As described in Section 3.2, the objective function of the SE InfoGAN is modified to enable D as an SNR indicator. Fig. 35 shows the value of the output scalar of D versus the SNR level of the input speech. Obviously, the output scalar of D tends to increase monotonically as the SNR gets higher. At 5 dB, the mean value of the output scalar is around 0.05, and eventually the mean value exceeds 0.5 at 35 dB SNR.

Interestingly, even though the SNR level for training is limited from -5 dB to 10 dB, it still works correctly when the SNR is higher than 10 dB. In fact, the mean value of the output scalar starts to increase rapidly when the SNR level is beyond 15 dB, which is out of the range of SNR levels used in training. The observation may imply that, while the terms $\mathbb{E}_{\mathbf{y}\sim clean}[(D(\mathbf{y})-1)^2]$ and $\mathbb{E}_{\mathbf{x}\sim noisy}[(D(\mathbf{x}))^2]$ could guide the D to distinguish different SNR levels of noisy speech, it struggles to tell which speech is more noisy especially at low SNR levels as the trend remains pretty flat, which is understandable as the D is not designed to be an accurate signal processing model that estimates precise SNR levels and the SNR-indicator only serves as an auxiliary function.

If one wishes to obtain a more precise SNR level, we suggest to specify the expected value of D's output scalar at different local SNR levels, which requires the metrics such as segmental SNR (SSNR) to be calculated over time. The regular SNR should not be used when defining the expected values, and the reason is that, while the SNR is typically precise when calculated over a long duration (i.e., a global metric), it is not an accurate short-time metric for unstationary noise whose spectrum changes over time, particularly when compared with the local SNR metric such as segmental SNR (SSNR). Consequently, as the noise dataset we normally used contains mostly unstationary noise, the functionality of D could not be learned very well through training, as the local SNR is subject to change over time while the global SNR remains

the same.

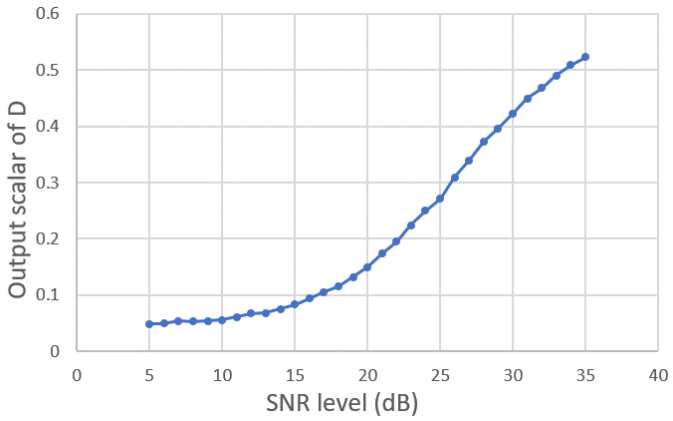


Figure 35: The value of output scalar of D at different SNR levels

Apparently, for high-SNR levels such as 35 dB, there is little need to actually process the noisy speech as the presence of noise is negligible, which makes it reasonable to skip the stage where clean speech is estimated. To further demonstrate how we could make use of the SNR-indicator functionality to save the system from redundant computation, we use a threshold $D_{thre} = 0.25$. Fig. 36 reveals the percentage of speech that has been processed with the threshold. It is apparent that the portion of speech being processed declines as the SNR increases, implying that the average computational cost is reduced since the G can be omitted during the speech-enhancing stage. For example, at 5 dB SNR, the percentage is about 100% and G is almost certain to be involved in the speech-enhancing stage, however, when the SNR is at 40 dB, the percentage decreases to approximately 5%, which means 95% of the processing does not require G.

As a metric of computational cost, we consider the average number of multiplies for obtaining 32 ms speech. For example, the mean number of multiplies of a model under Config. 2 reaches $39.8m$ in a noisy condition such as 0 dB SNR, where the percentage of speech being processed is nearly 100%, and it gradually decreases to $15.6m$ when there is little noise. On the other hand, the conventional GAN-based

SE method (FSEGAN) requires $207m$ multiplies to produce 32 ms speech under all SNR conditions, which is significantly larger than the proposed method. The model complexity, including the size of parameters and the averaged computational cost at 2 different SNR levels (5 dB and 40 dB) of all three models, is given in Table. 14. The computational cost of both CIRM and FSEGAN remains a constant at all SNR levels, while the proposed method scales the computations for different SNR levels. Interestingly, though the DNN-based CIRM model contains most parameters ($3.88m$), its computational cost remains a constant for all SNR levels and is actually the lowest among all three methods thanks to the simplicity of fully-connected structure that can be easily performed by matrix multiplication.

It should be noted that the complexity of the model totally depends on the configuration of the network, which is fairly flexible and could be tweaked in order to reduce the overall complexity. As mentioned, we do not attempt to find the best possible performance of each model due to the bias-variance tradeoff, but generally speaking, CNN-based GAN (such as FSEGAN and the proposed method) are usually more light-weighted than fully-connected-DNN-based methods (e.g., DNN-based CIRM) in terms of the number of parameters, while the latter often requires fewer computations.

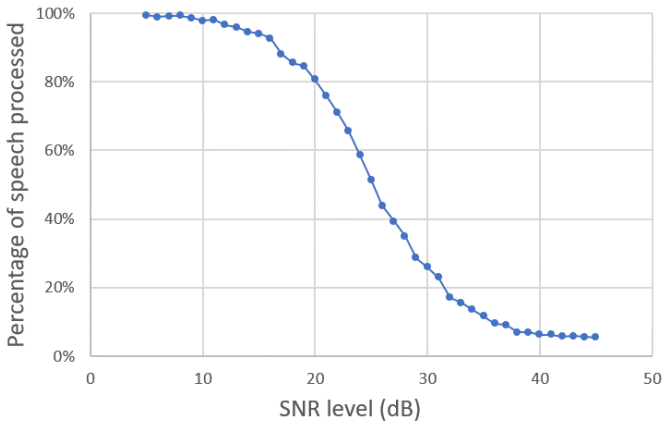


Figure 36: The percentage of speech processed by the system at different SNR levels

Table 14: The model complexity of different methods

	CIRM	**FSEGAN**	**Proposed**	
			config. 1	**config. 2**
Number of parameters	3.88m	1.46m	288k	**154.7k**
Number of multiplies (5 dB)	**3.88m**	207m	74m	39.8m
Number of multiplies (40 dB)	**3.88m**	207m	42m	15.6m

3.3.3 Performance measure of keyword spotting

In Section 2.3.2 we showed the impact of training dataset on KWS performance. In this section, however, we will investigate the improvement on KWS performance for different models. Through experiments, we wish to find a model that is suitable for KWS task, and the best possible performance that the model could achieve.

For evaluation, the same keyword spotting engine that detects keyword *Alexa* is used. Two metrics, namely false-reject rate (FRR) and false-alarm rate (FAR) are employed for evaluation. To measure FRR, we use a dataset that contains a stream of keyword *Alexa* at 5 SNR levels: -6 dB, -3 dB, 0 dB, 3 dB and 6 dB. For FAR evaluation, both the keyword dataset and a dataset that consists of 24-hour BBC recordings are used.

It should be mentioned that, there is usually a tradeoff between FRR and FAR performance. Reduction on FRR usually increases FAR and vise versa, because the system tends to be more *"sensitive"* towards keywords and is more likely to wrongly recognize other sound as the keyword. Since FRR is often considered to be more important than FAR, people are usually willing to sacrifice FAR performance for improving FRR.

For the proposed method, network structure is changed to the one illustrated in Fig.34 (b) as we found it outperforms the other. The number of parameters is similar

to that of FSEGAN, and the D and G now consist of $668k$ and $682k$ parameters, respectively. Table 15 shows the results on KWS of all three methods (CIRM, FSEGAN and the proposed) in terms of FRR. It is noticeable that, only FSEGAN can improve the KWS performance in all testing SNR scenarios while CIRM and the proposed method degrade the KWS performance at 6 dB SNR.

Table 15: Number of misses per 60 keywords on different models

	unprocessed	**CIRM**	**FSEGAN**	**Proposed**
-6 dB	56.2	46.4	**40.4**	49.3
-3 dB	46.8	37.3	**28.4**	38.2
0 dB	33.3	30.1	**17.0**	29.3
3 dB	21.3	20.6	**9.2**	21.3
6 dB	10.5	14.5	**5.0**	19.0

The corresponding FAR performance on keyword dataset and BBC recording dataset is given in Table 16 and Table 17, respectively. Overall, CIRM outperforms FSEGAN and the proposed method in terms of FAR. The proposed method produces a considerable number of FA on keyword dataset while the other two methods won't cause FA. On BBC recording dataset, on the other hand, FSEGAN leads to the highest FAR among all three methods.

As mentioned before, since FRR is more emphasized than FAR, consequently, the FSEGAN is strongly preferred over CIRM and the proposed method for KWS, even though CIRM achieves the best result in terms of FAR for both testing datasets. Interesting, while the proposed method performs very well in terms of speech intelligibility metrics, it falls short of the performance achieved by CIRM and FSEGAN.

FSEGAN is proven to be applicable towards SE for KWS task since it could effectively reduce the FRR while causing a considerable FAR in BBC-recording dataset. In order to further improve its performance on FAR, we incorporate the SNR-indicator functionality into the method by adopting the system framework which is shown in

Table 16: Number of false alarms per hour keyword recording on different models

	unprocessed	**CIRM**	**FSEGAN**	**Proposed**
-6 dB	0	0	0	1.5
-3 dB	0	0	0	1.1
0 dB	0	0	0	0.7
3 dB	0	0	0	7.2
6 dB	0	0	0	30.9

Table 17: Number of false alarms per hour BBC recording on different models

	FAR
unprocessed	0
CIRM	0
FSEGAN	0.83
Proposed	0.083

Fig. 33 and by modifying its objective function:

$$\min_D V(D) = \frac{1}{3}\mathbb{E}_{\mathbf{x}\sim noisy}[(D(\mathbf{x}))^2] + \frac{1}{3}\mathbb{E}_{\mathbf{y}\sim clean}[(D(\mathbf{y})-1)^2] + \frac{1}{3}\mathbb{E}_{\mathbf{x}\sim noisy}[(D(G(\mathbf{x})))^2] \tag{17}$$

$$\min_G V(G) = \mathbb{E}_{\mathbf{x}\sim noisy}[(D(G(\mathbf{x}))-1)^2] + \lambda\mathcal{L}_{\text{MSE}}(G(\mathbf{c}),\mathbf{y}) \tag{18}$$

Same as in Section 3.2, the extra term $\mathbb{E}_{\mathbf{x}\sim noisy}[(D(G(\mathbf{x})))^2]$ is added to the objective function. The FSEGAN is retrained with the new objective function and tested with the BBC recording dataset. Table 18 illustrates the FAR performance and the percentage of data processed with different D_{thre}. Obviously, the FAR could be effectively controlled as less data is processed.

In the meantime, setting a threshold to control the portion of speech being processed may impact the FRR performance. As shown in Table 19, the number of misses tends to increase by setting a lower threshold D_{thre}. However, by carefully choosing a value for D_{thre}, it is possible to avoid major impact on FRR while keeping

Table 18: Number of false alarms per hour BBC recording and the percentage of data processed under different D_{thre} on FSEGAN

	FAR	**% of data processed**
unprocessed	0	0%
$D_{thre} = 0.1$	0	15.8%
$D_{thre} = 0.2$	0.25	53.1%
$D_{thre} = 0.4$	0.833	97.6%

FAR at a relatively low level.

Table 19: Number of misses per 60 keywords with different D_{thre} on FSEGAN

	unprocessed	$D_{thre} = 0.1$	$D_{thre} = 0.2$	$D_{thre} = 0.4$
-6 dB	56.2	42.8	41.9	**40.3**
-3 dB	46.8	31.4	30.1	**28.4**
0 dB	33.3	20.7	18.3	**17.1**
3 dB	21.3	12.2	10.5	**9.3**
6 dB	10.5	7.2	5.4	**5.0**

3.4 Summary

In this chapter, a speech enhancement method is proposed based on information maximizing generative adversarial network. At first, some background materials on GAN, InfoGAN and some typical GAN-based speech enhancement methods are given. The proposed InfoGAN-based SE method, which incorporates the function of SNR indicator into the SE method, is then introduced. This functionality adds negligible additional cost to the system by adopting InfoGAN in a deterministic manner and employing the encoder design for discriminator and decoder design for generator, respectively. The overall model is capable of scaling its computational complexity

under different signal-to-noise ratio (SNR) levels by detecting clean or near-clean speech which requires no processing.

Through experiments, we have verified the functionality of the proposed method and demonstrated that the model can effectively improve the speech intelligibility. Further studies show that while the proposed method can improve the keyword spotting performance on noisy environments, conventional GAN-based SE method is still preferred for this task.

Chapter 4

Conclusion and Future Work

4.1 Summary of the work

In this thesis, deep-learning-based speech enhancement methods have been thoroughly studied. Specifically, neural network models including convolutional neural network and generative adversarial network have been investigated for single-channel speech enhancement task, with emphasis on both speech intelligibility and application to automatic speech recognition (ASR).

A fully convolutional neural network was first proposed in Chapter 2, which takes advantage of the WaveNet framework and features frequency-dilated 2d-convolution for complex spectrogram processing. WaveNet is a modern network that is proven to be applicable in modeling acoustic signal in time domain, yet cannot be simply applied to frequency-domain spectrogram estimation. The frequency-domain adaption of the WaveNet greatly reduces the time-complexity by replacing the autoregressive prediction with frequency-domain regression. We proposed the stacked frequency dilation to eliminate the need for increasing the size of filter while still enables the network to obtain a large receptive field. On the other hand, the complex spectrogram of speech is chosen as the input and output of the proposed network, which

implicitly processes both spectral phase and magnitude simultaneously. Through a series of experiments, we demonstrated that the proposed network performs fairly well even with limited number of parameters, and is applicable towards both speech intelligibility and application to keyword spotting system. By training the proposed model on different speech dataset, we showed the choice of training dataset could have a major impact on the final keyword spotting result.

In Chapter 3, we presented an InfoGAN-based method for single-channel speech enhancement. The conventional GAN-based speech enhancement models often adopt an encoder-decoder design for the generator and an encoder-classifier design for the discriminator, and the only the generator is employed in the speech-enhancing stage. By adapting the InfoGAN in a supervised, discriminative manner, the structure of generator is reduced to a single decoder and an auxiliary network is appended to the encoder-classifier structure of the discriminator, which result in a more compact model and a different speech-enhancing stage that involves both discriminator and generator. In addition, we have taken advantage of this speech enhancement process and modified the objective function to enable the discriminator to work like a SNR-indicator. Consequently, the system can scale the computational cost depending on the SNR level by detecting and omitting the clean or near-clean speech that requires no processing. The proposed model is able to yield a convincing performance as compared to the conventional GAN-based speech enhancement models in terms of speech intelligibility, while the conventional GAN-based model is preferred over the proposed model for keyword spotting task.

4.2 Suggestions for future work

The focus of this thesis is deep-learning-based single-channel speech enhancement methods. While the proposed methods are suitable for speech enhancement in reverberant environments, the training process has only used additive noise without considering room reverberations. It would be interesting to see the performance of the models in both noisy and reverberant conditions. In the meantime, although the single-channel speech enhancement is the objective of the proposed methods, most real-world application scenarios often provide the speech enhancement system with multi-channel speech. Since the proposed network models are fairly flexible and could be extended for multi-channel speech enhancement without major modifications, future work includes investigating the variations and performances of the proposed models for multi-channel speech enhancement.

In this thesis, we have evaluated the models at two aspects: speech intelligibility and ASR performance (on a keyword spotting engine). As illustrated in the experimental results, the keyword spotting performance is not always consistent with the intelligibility score. Nowadays, ASR has become a major application scenario for speech enhancement system, yet most of the conventional methods only focus on improving speech intelligibility, which are not applicable in the context of ASR. Hence a direction of future work can focus on minimizing the gap between speech intelligibility and the corresponding ASR performance.

Bibliography

[1] Librivox. [Online]. Available: https://librivox.org/

[2] Ted. [Online]. Available: https://www.ted.com/

[3] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3762–3769.

[4] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 716–720.

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in neural information processing systems*, 2016, pp. 2172–2180.

[6] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5024–5028.

[7] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[8] S.-W. Fu, T.-y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2017, pp. 1–6.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[10] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[11] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 55–66, 2015.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[14] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for

image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[16] ——, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] E. W. Healy, S. E. Yoho, J. Chen, Y. Wang, and D. Wang, "An algorithm to increase speech intelligibility for hearing-impaired listeners in novel segments of the same noise type," *The Journal of the Acoustical Society of America*, vol. 138, no. 3, pp. 1660–1669, 2015.

[18] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Tedlium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*. Springer, 2018, pp. 198–208.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[20] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2009–2022, 2016.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[23] M. Krawczyk and T. Gerkmann, "STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1931–1940, Dec 2014.

[24] A. A. Kressner, T. May, and C. J. Rozell, "Outcome measures based on classification performance fail to predict the intelligibility of binary-masked speech," *The Journal of the Acoustical Society of America*, vol. 139, no. 6, pp. 3033–3036, 2016.

[25] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[26] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12. IEEE, 1987, pp. 705–708.

[27] B. Liu, S. Nie, Y. Zhang, D. Ke, S. Liang, and W. Liu, "Boosting noise robustness of acoustic model via deep adversarial training," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5034–5038.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[29] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[30] D. Michelsanti and Z.-H. Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," *arXiv preprint arXiv:1709.01703*, 2017.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[32] Z. Ouyang, H. Yu, W.-P. Zhu, and B. Champagne, "A deep neural network based harmonic noise model for speech enhancement," *Proc. Interspeech 2018*, pp. 3224–3228, 2018.

[33] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465–494, 2011.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[35] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," *arXiv preprint arXiv:1609.07132*, 2016.

[36] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[37] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[38] ——, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1015–1018.

[39] H. Purwins, B. Li, T. Virtanen, J. Schlter, S. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, May 2019.

[40] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[41] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5069–5073.

[42] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[44] A. Rousseau, P. Deléglise, and Y. Esteve, "Ted-lium: an automatic speech recognition dedicated corpus." in *LREC*, 2012, pp. 125–129.

[45] ——, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks." in *LREC*, 2014, pp. 3935–3939.

[46] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *16th Annual Conference of the International Speech Communication Association*, 2015.

[47] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[48] L. Sun, J. Du, L.-R. Dai, and C.-H. Lee, "Multiple-target deep learning for lstm-rnn based speech enhancement," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*. IEEE, 2017, pp. 136–140.

[49] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[50] K. Tan, J. Chen, and D. Wang, "Gated residual networks with dilated convolutions for supervised speech separation," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 21–25.

[51] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[52] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with pixelcnn decoders," in *Advances in neural information processing systems*, 2016, pp. 4790–4798.

[53] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[54] J. Long, E. Shelhamer, and T. Darrell. A fuller understanding of fully convolutional networks. [Online]. Available: http://www.micc.unifi.it/bagdanov/pdfs/FCN-presentation.pdf

[55] D. Wang and J. Chen, "Supervised speech separation based on deep learning:

An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[56] K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, and L. Xie, "Investigating generative adversarial networks based speech dereverberation for robust speech recognition," *arXiv preprint arXiv:1803.10132*, 2018.

[57] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, Dec 2014.

[58] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, March 2016.

[59] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, Jan 2015.

[60] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5485–5493.

[61] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015. [Online]. Available: http://arxiv.org/abs/1511.07122